

GESS ChartFactory

Anwenderhandbuch



Gesellschaft für Software
in der Sozialforschung mbH

Waterloohain 6 – 8
22769 Hamburg
Tel.: 040 - 853 753 - 0
Fax: 040 - 853 753 - 33
www.gessgroup.de

GESS ChartFactory
Anwenderhandbuch

Knut Holzcheck, Ercan Nak, Judith Krüger

Stand: November 2019

GESS Gesellschaft für Software in der Sozialforschung mbH
Waterloohain 6-8
22769 Hamburg
Tel. 040 - 853 753 0
Fax 040 - 853 753 433
software@gessgroup.de
www.gessgroup.de

Inhaltsverzeichnis

1 Grundlagen	1
1.1 Lizenziieren	1
1.2 Technische Voraussetzungen	2
1.3 GESS Produktionsstraße	2
1.4 Nötige Einstellungen im Skript	3
1.5 Starten des Charting-Prozesses	3
2 Vorbereitung der PowerPoint-Vorlage	4
3 Einführung in das Charting	8
3.1 Beispiel 1: einfaches Diagramm	8
3.2 Beispiel 2: INVERSE	10
3.3 Beispiel 3: Basen im Labeltext	11
3.4 Beispiel 4: Template p	12
3.5 Beispiel 5: einfaches Liniendiagramm	15
3.6 Beispiel 6: START	17
3.7 Beispiel 7: DIRECTION FALLING	21
4 Texte	25
4.1 In Diagrammen	25
4.2 Textboxen	25
4.3 Tabellen	28
5 Liniengrafiken	33
5.1 Liniendiagramme	33
5.2 Profildiagramme	34
5.3 GESS-Profildiagramm	38
6 Einbindung von Makros	41
7 POWERCHARTOPTION	46
7.1 Markstring	47
7.2 Transpon	47
7.3 LabelIconReplace	48
7.4 XYScatterIcon	50
7.5 BaseCheck	51
7.6 Sort	54
7.7 ValueDiffToMax/ ValueDiffToMin	55
7.8 Formula	56
7.9 DataFromGraph	58
7.10 AxisPri	58
7.11 CellColor	59
7.12 CellFontColor	60
7.13 CellIcon	61
7.14 PerRow/ PerCol	62
7.15 SeriesColor	63
7.16 SeriesLabelColor	64
7.17 SeriesColorMarkstring	65
7.18 SeriesLabelAdd	67

7.19 SeriesLabelReplace	67
7.20 FillColorObject	68
7.21 DeleteObject	69
8 Daten-Aktualisierungen	71
9 GESStabs-spezifische Hinweise	73
9.1 Tabelleninhalte	73
9.2 Leere Tabellenzellen	73
9.3 Mehrseitige Tabellen	73
9.4 Formatierung von Texten	74
9.5 Multiuser	74
9.6 Berechnete Tabellenelemente	74
10 PowerPoint-spezifische Hinweise	75
10.1 Kontrolle von Formatierungen	75
10.2 Diagramm-interne Beschriftungen	75
10.3 Skalierung einzufügender Bilder	75
10.4 Ausblenden von Nullen in Diagrammen	76
10.5 'AutoWiederherstellen'-Funktion	77
11 Syntax	79
11.1 Basisbefehl	79
11.2 Optionale Befehlsbestandteile	81
11.3 Einfache Übernahme von Seiten	85
11.4 Zusätzliche Befehle	85
12 Syntax - Verzeichnis	87
Stichwortverzeichnis	92

Abbildungsverzeichnis

1.1	Arbeitsweise der GESS ChartFactory	1
1.2	Einstellung der Makro-Sicherheit in MS Excel	2
2.1	PowerPoint-Auswahlbereich	4
2.2	GESS-PowerPoint-Master	5
2.3	Elemente im GESS-PowerPoint-Master	5
2.4	Standardbenennung von Elementen durch PowerPoint	6
2.5	PP-Elemente anhand von Schlüsselwörtern umbenennen	6
3.1	Vorlage für ein einfaches Balkendiagramm	8
3.2	Ausgabe für ein einfaches Balkendiagramm	9
3.3	Markierte Datenblatt-Bereiche	10
3.4	Ausgabe mit 'INVERSE'	11
3.5	In Labeltexte integrierte Basen-Angaben	12
3.6	Vorlagechart mit mehreren Hauptelementen	13
3.7	Ausgabechart mit mehreren Hauptelementen	15
3.8	Vorlage für ein einfaches Liniendiagramm	16
3.9	Ausgabe für ein einfaches Liniendiagramm	17
3.10	Vorlage für ein Kategoriendiagramm	18
3.11	Datenblatt des Kategoriendiagrammes	19
3.12	Ausgabe mit Kategoriendiagramm	21
3.13	Chartvorlage mit positiv und negativ gerichteten Werten	22
3.14	Datentabelle mit positiv und negativ gerichteten Werten	22
3.15	Ausgabechart mit 'DIRECTION FALLING'	24
4.1	Diagramm mit integrierten Labeltexten	25
4.2	Vorlage mit Textboxen	26
4.3	Labeltexte in Textboxen	26
4.4	Gestaltungsmöglichkeiten von Textboxen	27
4.5	Vorlage mit Texttabelle	28
4.6	Labeltexte in Texttabelle	29
4.7	Textformatierung mit 'MARKSTRING'	30
4.8	Bilder auf erster Seite der Vorlagedatei	31
4.9	Textersatz mit POWERCHARTOPTION 'LabelIconReplace'	32
5.1	Vorlage für Liniendiagramm	33
5.2	Datenblatt eines Liniendiagramms	34
5.3	Ausgabe mit Liniendiagramm	34
5.4	Vorlage für Profildiagramm	35
5.5	Datenblatt eines Profildiagramms	35
5.6	Ausgabe mit Profildiagramm	37
5.7	Vorlage für GESS-Profildiagramm	38
5.8	Datenblatt des GESS-Profildiagramms	38
5.9	Ausgabe mit GESS-Profildiagramm	39
5.10	Befülltes Datenblatt des GESS-Profildiagramms	39
5.11	Festlegung eines Achsenbereichs mit POWERCHARTOPTION 'AxisPri'	40
6.1	Vorlagechart für Charting-Makro	41
6.2	Ausgabe der mit einem Makro erzeugten Charts	43

6.3	Vorlage für Makro 'autochart'	45
6.4	Ausgabe mit Makro 'autochart'	45
7.1	POWERCHARTOPTION 'Transpon (GESStabs-Tabelle)'	48
7.2	POWERCHARTOPTION 'Transpon'	48
7.3	Bilder auf erster Seite der Vorlagedatei	49
7.4	Vorlage für LabelIconReplace	50
7.5	LabelIconReplace	50
7.6	POWERCHARTOPTION 'XYScatterIconreplace'	51
7.7	Varianten von POWERCHARTOPTION 'BaseCheck'	53
7.8	Weitere Varianten von POWERCHARTOPTION 'BaseCheck'	54
7.9	POWERCHARTOPTION 'Sort'	55
7.10	POWERCHARTOPTION 'ValueDiffToMax/ Min'	56
7.11	POWERCHARTOPTION 'Formula'	57
7.12	POWERCHARTOPTION 'DataFromGraph'	58
7.13	POWERCHARTOPTION 'AxisPri' mit Warnhinweis	59
7.14	POWERCHARTOPTION 'CellColor'	60
7.15	POWERCHARTOPTION 'Cellfontcolor'	61
7.16	POWERCHARTOPTION 'Cellicon'	62
7.17	POWERCHARTOPTION 'Cellcolor PerRow'	63
7.18	POWERCHARTOPTION 'Seriescolor'	64
7.19	POWERCHARTOPTION 'SeriesLabelcolor'	65
7.20	POWERCHARTOPTION 'SeriesColorMarkstring'	66
7.21	POWERCHARTOPTION 'SeriescolorMarkstring'	66
7.22	POWERCHARTOPTION 'SeriesLabelAdd'	67
7.23	POWERCHARTOPTION 'SeriesLabelReplace'	68
7.24	Vorlage 1 für POWERCHARTOPTION 'FillColorObject'	69
7.25	POWERCHARTOPTION 'FillColorObject' 1	69
7.26	Vorlage 2 für POWERCHARTOPTION 'FillColorObject'	69
7.27	POWERCHARTOPTION 'FillColorObject' 2	69
7.28	Vorlage für POWERCHARTOPTION 'DeleteObject'	70
7.29	POWERCHARTOPTION 'DeleteObject'	70
8.1	Manuell ergänztes Chart	71
8.2	Mit 'POWERCHARTREIMPORT' aktualisiertes Chart	72
9.1	'FONT' im Tabellentext	74
9.2	'FONT' im Diagramm	74
10.1	Deaktivieren von 'Seitenverhältnis sperren'	76
10.2	Ausblenden von Nullen	77
10.3	'AutoWiederherstellen'-Funktion	78

Tabellenverzeichnis

3.1	Mit GESStabs erstellte Tabelle	8
3.2	Mittelwerte-Tabelle	15
3.3	Bewertung der Automarken nach Kategorien	18
3.4	Zufriedenheit mit Automarken, Gruppe 4	21
5.1	Mittelwerte-Tabelle	33
6.1	Zufriedenheit mit Automarken, Gruppe 1	41
6.2	Tabelle für Makro 'autochart'	45
7.1	Anwendungsgebiete der POWERCHARTOPTIONS	46
7.2	Wildcards in Suchtexten	47
7.3	Optionen für POWERCHARTOPTION 'LabelIconReplace'	49
7.4	Optionen für POWERCHARTOPTION 'XYScatterIcon'	50
7.5	Definition des Wertebereichs für POWERCHARTOPTION 'BaseCheck'	51
7.6	Optionen für POWERCHARTOPTION 'BaseCheck'	52
7.7	Referenzangaben bei unbekannter Tabellengröße für POWERCHARTOPTION 'Formula'	57
7.8	Formulierung von Bedingungen für POWERCHARTOPTION 'CellColor'	60
7.9	Optionen für POWERCHARTOPTION 'Cellicon'	61

1 Grundlagen

GESS ChartFactory ist eine ergänzende Anwendung zu GESStabs: Inhalte aus Datentabellen, die mit GESStabs erstellt wurden, werden automatisiert in eine PowerPoint-Datei übertragen.

Grundlage einer jeden Chart-Erstellung mittels GESS ChartFactory ist ein TABLE-Befehl, der eine Tabelle generiert und mit Daten sowie Texten befüllt. Außerdem wird eine PowerPoint-Datei mit Vorlage-Charts benötigt – diese bildet die Layout-Grundlage einer jeden Chart-Erzeugung und kann vom Benutzer frei gestaltet werden.

Im Skript wird mit **POWERCHART** die Anfertigung eines Charts in PowerPoint angefordert und anschließend definiert, mit welchen Informationen aus der Datentabelle die Elemente des Vorlage-Charts befüllt werden sollen. Übertragungsdateien fassen die Angaben zusammen, übertragen und speichern diese in einer PowerPoint-Ausgabe-Datei.

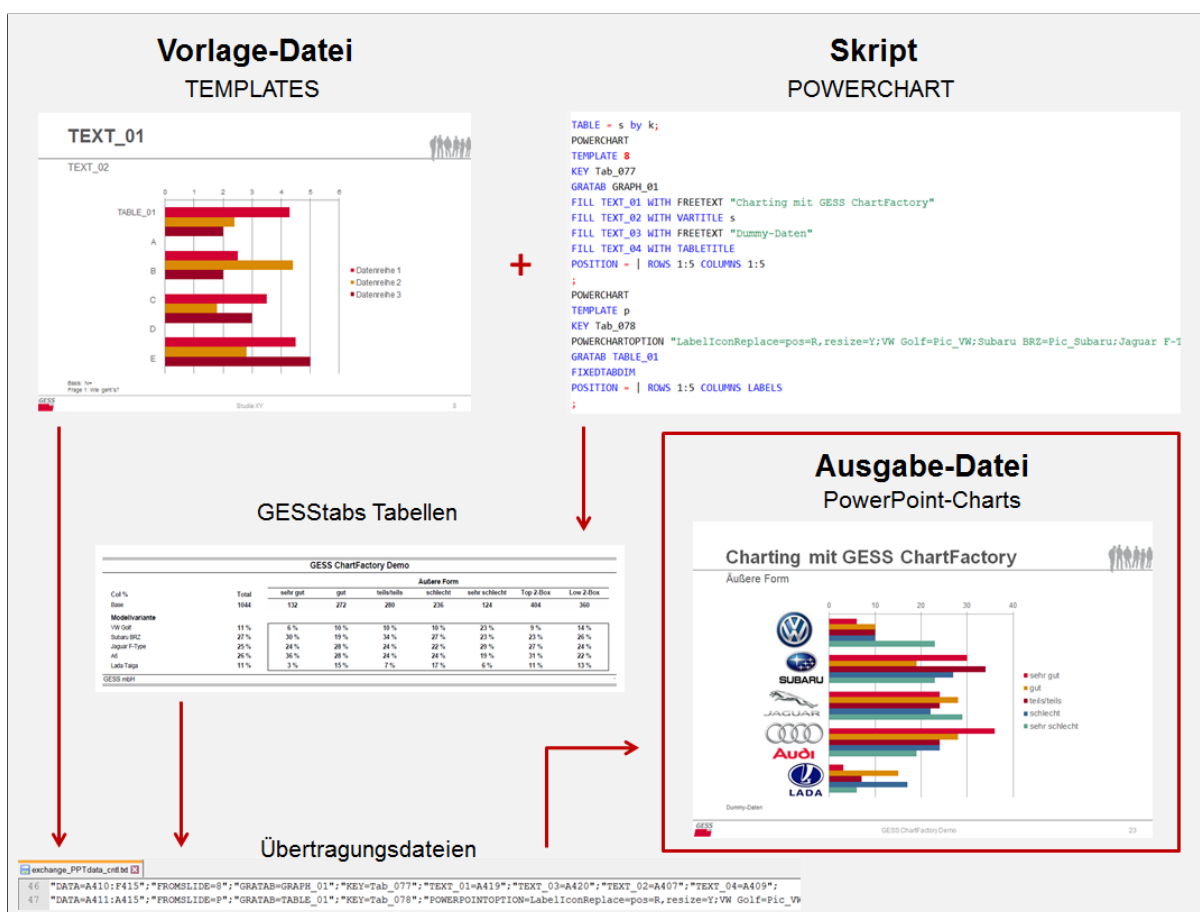


Abb. 1.1: Arbeitsweise der GESS ChartFactory

1.1 Lizenzieren

Zur Lizenzierung der GESS ChartFactory wird die heruntergeladene Powercharts.xlsb-Datei geöffnet und der personalisierte Lizenzschlüssel in die Zelle C1 eingetragen. Anschließend wird die Datei gespeichert und im GESStabs-Arbeitsverzeichnis abgelegt.

1.2 Technische Voraussetzungen

Für die Chart-Erzeugung werden – neben den von GESS vertriebenen Produkten GESS ChartFactory und GESStabs – externe Software-Produkte benötigt, die eigenen Lizenzen unterliegen: Microsoft Excel und Microsoft PowerPoint (ab Version 2007).

GESStabs übersetzt die Chart-Anforderungen des POWERCHART-Befehls in ein internes Datenformat (Übertragungsdateien), das von den Microsoft-Produkten interpretiert und in PowerPoint-Charts umgewandelt wird. Dieser Ablauf wird von GESStabs automatisch durchgeführt.

Seit November 2016 wird die GESS ChartFactory in der Entwicklung als XLSB-Datei fortgeführt. Damit Excel die Powercharts.xlsb ausführen kann, muss die Makro-Sicherheit innerhalb des Programms auf die 'unterste Stufe' gestellt werden.

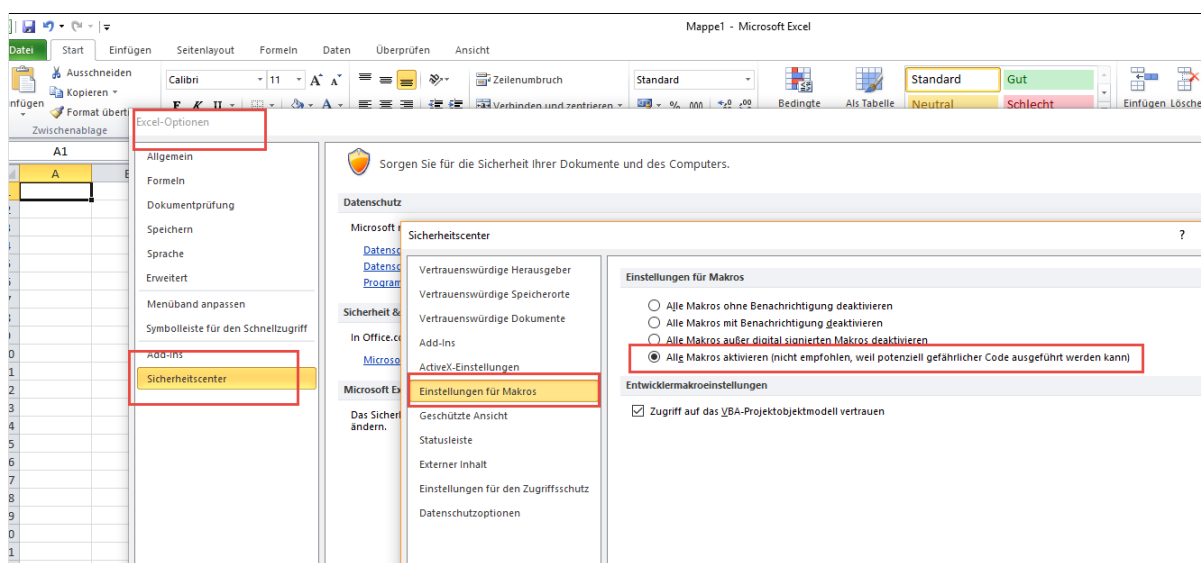


Abb. 1.2: Einstellung der Makro-Sicherheit in MS Excel

Bemerke außerdem: Die GESS-Software deaktiviert die 'AutoWiederherstellen'-Funktion der Microsoft-Software, wodurch ein stabilerer und schnellerer Software-Ablauf erreicht wird¹. Wir empfehlen daher ein regelmäßiges Zwischenspeichern der Arbeitsdateien, um Verluste von Daten und Arbeitsständen zu vermeiden.

1.3 GESS Produktionsstraße

Zusätzlich zur Datenerhebung mit GESS Cati und GESS Q., der anschließenden Tabellierung mit GESStabs und der Ausgabe der Tabellenbände in PDF, Excel, Postscript und HTML bietet GESS mit der GESS ChartFactory eine Option zur automatischen Erzeugung von Chartbänden in PowerPoint- und PDF-Dateiformat.

¹Wenn gewünscht, kann diese Funktion wieder aktiviert werden, siehe 'AutoWiederherstellen'-Funktion.

1.4 Nötige Einstellungen im Skript

Damit GESStabs die Powercharts.xlsb ausführt - und nicht, wie zuvor, die Powerchart.exe - , ist im Skript folgende Voreinstellung nötig:

```
PowerchartUseExe = no;
```

Für ein lauffähiges Charting-Projekt müssen außerdem die folgenden Dateien in das Skript eingebunden bzw. benannt werden:

1. Definition der Vorlage-Datei

```
PPTEMPLATES = "PP-Vorlage.pptx";
```

Name der Datei, die die Chartvorlagen enthält, oder Angabe des Dateipfades (falls diese Datei nicht im selben Dateordner liegt wie das Skript).

2. Benennung der Ausgabedatei

```
POWERPOINTFILENAME = "PP-Ausgabe.pptx";
```

Benennung der Datei, in die die Ergebnischarts geschrieben werden, oder Angabe des Dateipfades (falls diese Datei nicht in denselben Dateordner geschrieben werden soll wie das Skript).

3. Optionale Benennung der Übertragungsdateien

```
PPEXCHANGE = "PPT_data";
```

Beim Starten des POWERCHART-Statements werden die Tabelleninformationen über zwei Text- sowie eine Excel-Datei an die PowerPoint-Generierungssoftware übertragen. Diese Dateien haben einen gemeinsamen Dateinamen-Prefix (exchange_), dann wird der frei wählbare Dateinamen-Teil (hier: PPT) eingetragen, worauf die jeweilige Datei-Kennung (_cntl.txt, _data.txt und _data.xlsx) folgt. Wird kein Dateinamen-Teil definiert, erzeugt die Software einen Zufallsnamen, der diesen Suffixen vorangestellt wird.

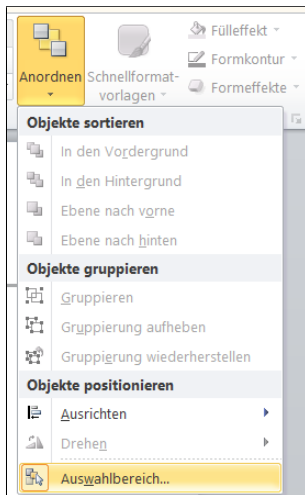
1.5 Starten des Charting-Prozesses

1. Schließen aller Microsoft Office-Anwendungen
2. Starten des GESStabs-Skripts
3. Excel und PowerPoint öffnen sich automatisch, Übertragungsdateien sowie Ausgabe-PowerPoint-Präsentation werden erzeugt, gespeichert und alle Programme werden wieder beendet

2 Vorbereitung der PowerPoint-Vorlage

Die Vorlage-Datei (TEMPLATE) enthält Chart-Vorlagen, die anschließend mit Informationen aus den GESSStabs-Tabellen befüllt werden. Sie dient als Sammlung aller gewünschten Charttypen und wird nicht überschrieben.

Die grundlegende Funktion, die für die Arbeit mit der PowerPoint-Vorlage benötigt wird, ist der **Auswahlbereich** am rechten Bildschirmrand, der alle Elemente auf dem angezeigten Chart und deren Ebenen-Stellung anzeigt.



Dieser befindet sich im Menü unter
Start → **Anordnen** → **Auswahlbereich** .

Abb. 2.1: PowerPoint-
Auswahlbereich

Das Layout und die Inhalte der Template legt der Benutzer selbst fest.

Die erste Seite einer jeden Vorlage-Datei ist für Bilder und Icons reserviert. Hier können beliebig viele Abbildungen gespeichert werden, die ggf. in der PowerPoint-Präsentation verwendet und im Zuge des automatisierten Chartings in die Charts eingefügt werden sollen¹.

Ab Seite 2 der Vorlage-Datei folgen die Roh-Charts, die mit Daten und Texten befüllt werden sollen.

¹Siehe hierzu das Unterkapitel zur POWERCHARTOPTION [LabelIconReplace](#).

Ein (unbefülltes) **Vorlage-Chart** sieht beispielsweise so aus:



Abb. 2.2: GESS-PowerPoint-Master

Hier sind bereits einige grundlegende Folienelemente eingerichtet und benannt worden: Titel der Folie, Platz für Kommentierungen, eine Fußnote für Basenangaben und Fragetexte, der Studientitel, das aktuelle Datum und eine Seitenzahl.

Die rechts stehende Abbildung zeigt die Benennung und Zuordnung der einzelnen Elemente. Wie zu erkennen ist, erfolgt diese anhand von festgelegten **Schlüsselwörtern**.

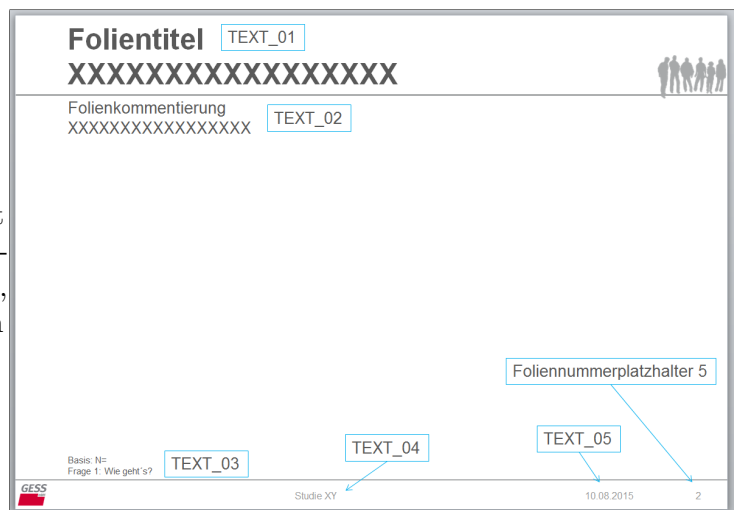


Abb. 2.3: Elemente im GESS-PowerPoint-Master

Die Schlüsselwörter lauten:

TEXT_[xx] für Textfelder

GRAPH_[xx] für Grafiken und Diagramme

TABLE_[xx] für Tabellen (können mit Zahlen und/ oder Texten befüllt werden)

[xx] steht für eine zweistellige Nummerierung zwischen 01 und 99.

Die Benennung von GRAPH- und TABLE-Elemente kann zum Zwecke einer inhaltlichen Kennung außerdem durch einen beliebigen Text erweitert werden, etwa: GRAPH_01Balken.

Wichtig: Die Elemente müssen zwingend nach diesem Schema benannt werden, um sie mit POWERCHART ansteuern und mit Inhalten füllen zu können².

²Der 'Foliennummerplatzhalter 5' wird in diesem Fall nicht nach diesem Schema benannt, weil es sich dabei um eine Konstante (Seitenzahl) handelt, die von PowerPoint automatisch befüllt wird.

Möchte man dem Chart nun neue Elemente hinzufügen, beispielsweise ein einfaches Balkendiagramm, erhalten diese zunächst eine Standardbenennung durch PowerPoint, hier: 'Diagramm 21'.

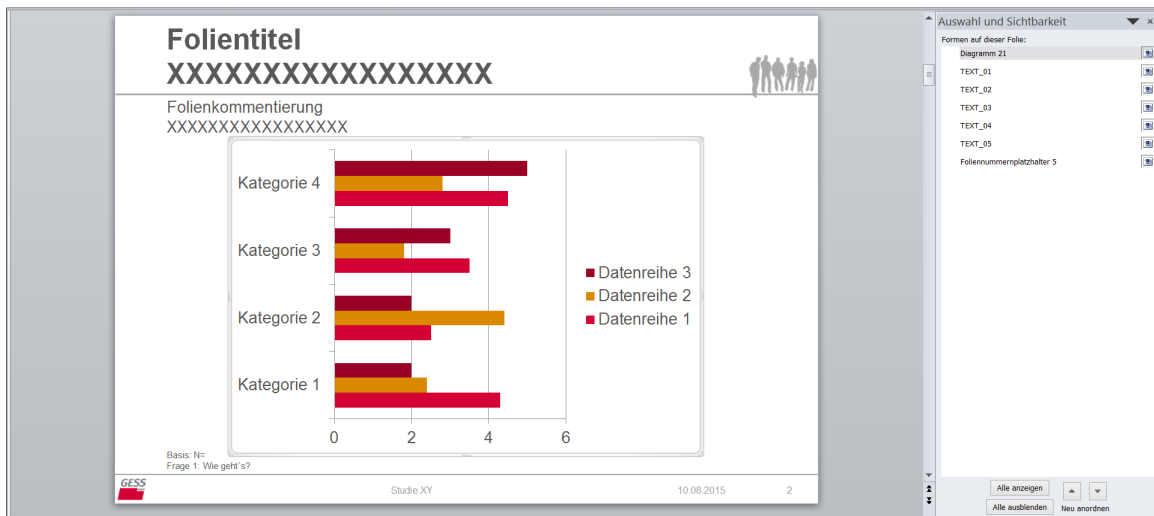


Abb. 2.4: Standardbenennung von Elementen durch PowerPoint

Diese müssen im Auswahlbereich geändert werden (auf den Namen klicken und editieren), um das Element mit POWERCHART ansprechen zu können: Da es das erstes Diagramm auf diesem Chart ist, wird es GRAPH_01 genannt.

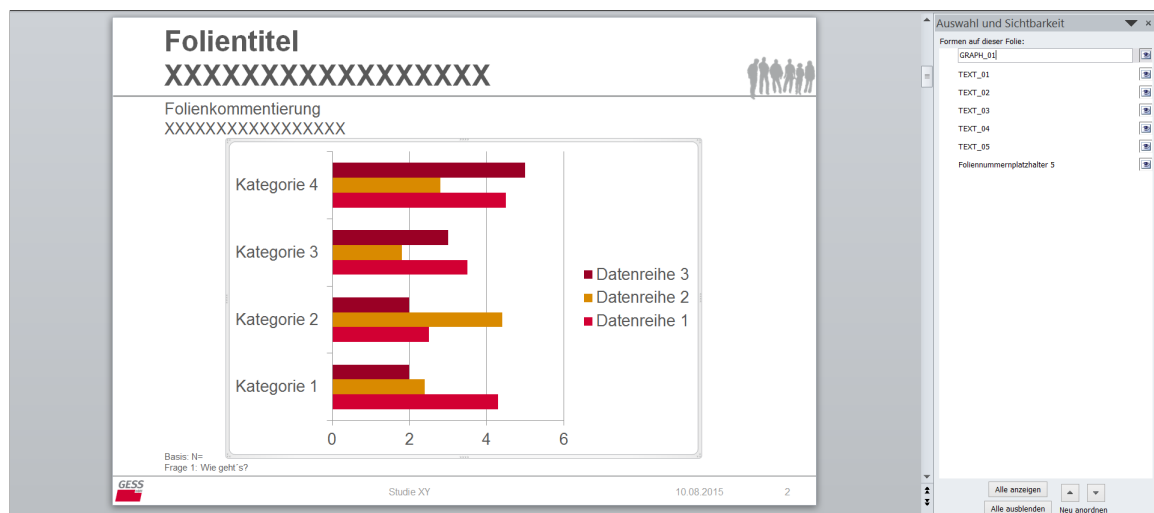


Abb. 2.5: PP-Elemente anhand von Schlüsselwörtern umbenennen

So können diesem und weiteren Vorlagecharts nun nach und nach Elemente bzw. Platzhalter hinzugefügt werden, die — entsprechend benannt — mit Inhalten gefüllt werden können.

Noch ein Hinweis: Die Elemente müssen innerhalb eines Charts eindeutig benannt werden. Auf dem nächsten Vorlagechart kann wieder mit bspw. GRAPH_01 begonnen werden. Da die Identifizierung der Elemente anhand der Kombination aus Chart und Element erfolgt, wird GRAPH_01 auf einem Chart nicht mit dem gleichbenannten Element auf einem anderen Chart verwechselt.

Und noch ein Hinweis: GESS ChartFactory passt Grafiken und Tabellen automatisch an die Beschaffenheit der Daten an. Wie viele Kategorien und Datenreihen das Vorlageelement abbildet

und wie dieses skaliert ist, spielt keine Rolle — gibt es mehr oder weniger darzustellende Elemente, wird die Darstellung im Ergebnischart angepasst und der vorhandene Bereich entsprechend ausgenutzt. Ebenso erfolgt eine Anpassung des darzustellenden Skalenbereiches an die Spannweite der Daten³. Zur generellen Kontrolle von Formatierungen und Diagramm- sowie Tabelleneinstellungen empfehlen wir, in den Vorlageelementen die maximale Anzahl an Datenreihen einzurichten und wie gewünscht zu formatieren⁴.

Die vorbereitete Vorlage-Datei wird nun unter einem beliebigen Namen (hier 'PP-Vorlage.pptx') im Arbeitsverzeichnis abgelegt und mit `PPTEMPLATES = "PP-Vorlage.pptx";` im Skript aufgerufen (siehe Kapitel: [Nötige Einstellungen im Skript](#)).

³Wird keine automatische Anpassung durch PowerPoint gewünscht, können Diagramm- und Tabelleneigenschaften wie Größe, Datenrange, Skalierung und Formatierung natürlich fixiert und gesteuert werden. Siehe dazu beispielsweise die Funktionen `FIXEDTABDIM` und `FIXEDRANGE` im [Syntax](#)-Kapitel und die `POWERCHARTOPTION` „AxisPri“

⁴Siehe hierzu auch [Kontrolle von Formatierungen](#).

3 Einführung in das Charting

3.1 Beispiel 1: einfaches Diagramm

Für den Einstieg in das Charting mit GESS ChartFactory wird von einer einfachen Kreuzta-
belle ausgegangen, die die Beurteilung der äußeren Form verschiedener Automodelle auf einer
fünfstelligen Likert-Skala zeigt.

Erstellt wurde diese mit dem einfachen GESStabs-Befehl: `TABLE = k BY s;`¹.

Tab. 3.1: Mit GESStabs erstellte Tabelle

GESS ChartFactory Demo					
Col %	Modellvariante				
	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
Base	116	280	264	272	112
Äußere Form					
sehr gut	7 %	14 %	12 %	18 %	4 %
gut	24 %	19 %	29 %	28 %	36 %
teils/teils	24 %	34 %	26 %	25 %	18 %
schlecht	21 %	23 %	20 %	21 %	36 %
sehr schlecht	24 %	10 %	14 %	9 %	7 %
Top 2-Box	31 %	33 %	41 %	46 %	39 %
Low 2-Box	45 %	33 %	33 %	29 %	43 %

GESS mbH

Diese soll in PowerPoint in einem einfachen Balkendiagramm dargestellt werden (zunächst einmal
ohne die unten stehenden Overcodes).

In die PowerPoint-Vorlagedatei fügt man dafür auf einem — abgesehen von den GESS-
Basiskomponenten — leeren Chart ein Balkendiagramm der Wahl ein und benennt es in GRAPH_01
um:

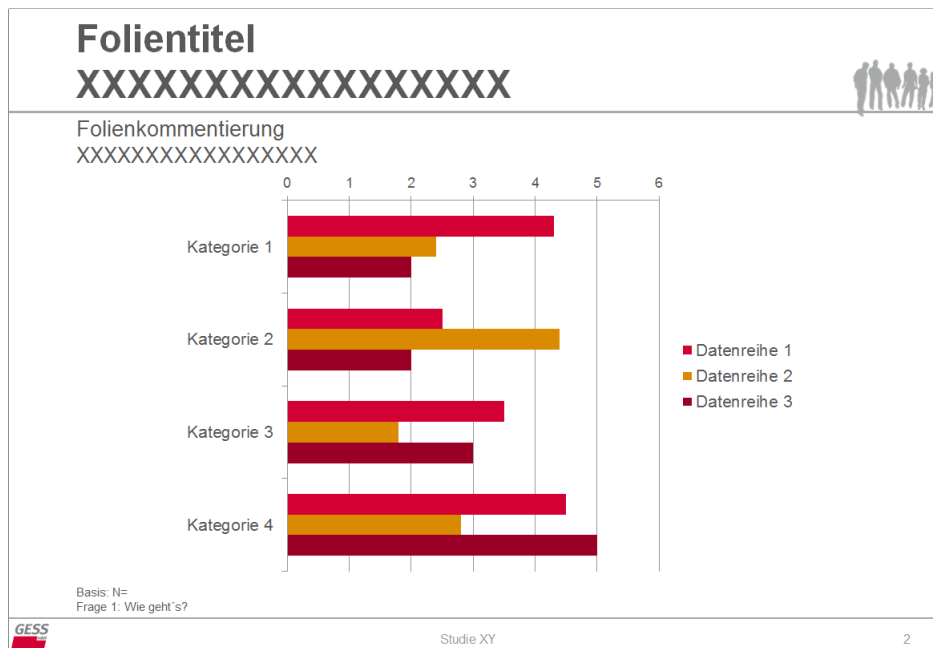


Abb. 3.1: Vorlage für ein einfaches Balkendiagramm

¹Variable k enthält das Automodell, s ist die Beurteilung der äußeren Form auf einer fünfstufigen Likert-Skala.

Dann formuliert man den POWERCHART-Befehl:

```
POWERCHART
TEMPLATE 2
KEY Tab_001
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT (VARTITLE s)
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION = | ROWS 1:5 COLUMNS 1:5
;
```

Mit POWERCHART startet man den Charting-Befehl. TEMPLATE 2 steuert die zweite Seite der Vorlagendatei an und selektiert dieses als Vorlagechart, KEY Tab_001 gibt dem Befehl die Kennung Tab_001. Dann wird bestimmt, dass das Diagramm namens GRAPH_01 angesprochen werden soll (GRATAB GRAPH_01). Quasi nebenbei befüllt man die Textelemente des GESS-Vorlagecharts (FILL TEXT_[xx] WITH CONCAT) mit Elementen aus der Tabelle (VARTITLE s und TABLETITLE) sowie frei bestimmbar Texten (in Anführungszeichen). POSITION = | ROWS 1:5 COLUMNS 1:5 gibt an, dass die erste bis fünfte Zeile sowie Spalte 1 bis 5 auf dem Chart dargestellt werden sollen². Das Statement wird mit einem Semikolon abgeschlossen.

Das Ergebnis sieht so aus:

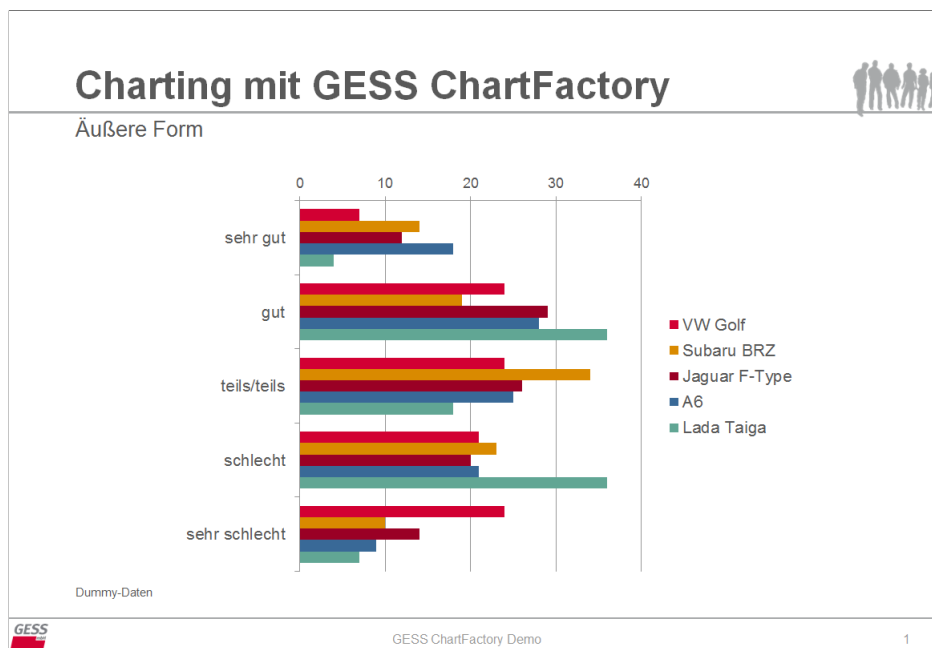


Abb. 3.2: Ausgabe für ein einfaches Balkendiagramm

Wie man sieht, hat sich das Diagramm, das ursprünglich auf die Abbildung von vier Items und drei Datenreihen ausgelegt und beliebig skaliert war, automatisch an die entsprechende Anzahl der Variablen-Kategorien angepasst, eine passende Skalierung gewählt und optisch ausgerichtet.

²Zur Auswahl des zu verwendenden Tabellenbereiches wird hier direkt mit dem POSITION-Statement eingestiegen. Eine andere Variante kommt ohne das POSITION-Schlüsselwort aus und benennt direkt die zu verwendenden Labelcodes (siehe dazu: [Syntax](#)). Dies stößt aber an einigen Stellen an ihre Grenzen – mit der hier verwendeten Variante kann nichts falsch gemacht werden!

Ein Blick in das Datenblatt des Diagramms zeigt, dass die Datenbereiche, die mit GESS ChartFactory befüllt wurden, mit gelbem Zellhintergrund gekennzeichnet wurden.

	A	B	C	D	E	F	
1	Spalte1	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga	
2	sehr gut	7	14	12	18	4	
3	gut	24	19	29	28	36	
4	teils/teils	24	34	26	25	18	
5	schlecht	21	23	20	21	36	
6	sehr schlecht	24	10	14	9	7	
7							

Abb. 3.3: Markierte Datenblatt-Bereiche

Dies ist insbesondere nützlich, wenn das Diagramm in der Chartvorlage mehr Datenreihen enthält, als letztendlich eingefügt werden³.

3.2 Beispiel 2: INVERSE

Nachvollziehbar wäre nun der Wunsch, anstatt der jetzigen gebündelten Darstellung der Bewertungskategorien die einzelnen Automodelle auf der X-Achse und ihre jeweilige Verteilung über die 5er-Skala abzubilden. In PowerPoint ist dies beispielsweise mit der Funktion 'Zeile/Spalte wechseln' machbar, in der GESS ChartFactory gibt es hierfür die einfache Anweisung INVERSE, mit der die Achsen-Darstellung gedreht wird. Das Schlagwort wird einfach in den Befehl eingefügt:

```
POWERCHART
TEMPLATE 2
KEY Tab_002
GRATAB GRAPH_01
INVERSE
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT (VARTITLE s)
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION = | ROWS 1:5 COLUMNS 1:5
;
```

³Im Ergebnisdiagramm werden dann zwar nur die eingefügten Datenreihen dargestellt, das Datenblatt wird jedoch nicht automatisch 'bereinigt' und enthält ggf. noch (Dummy-)Daten des Vorlagediagramms.

Das Diagramm sieht dann wie folgt aus:

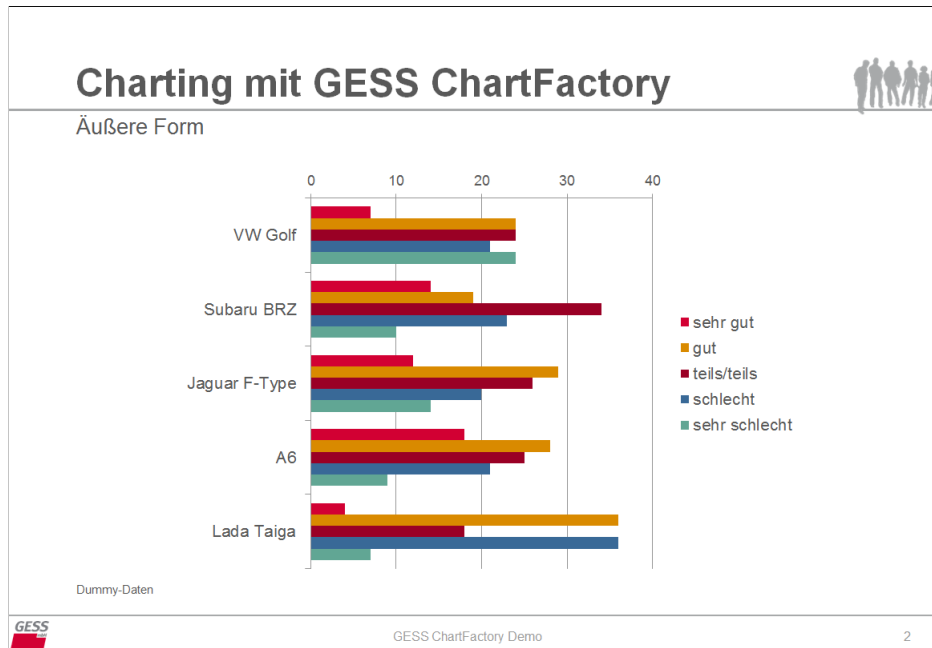


Abb. 3.4: Ausgabe mit 'INVERSE'

3.3 Beispiel 3: Basen im Labeltext

Es ist üblich, auf PowerPoint-Charts die abgebildeten Daten durch die zugrunde liegenden Fallzahlen zu ergänzen. Eine Möglichkeit, Basenangaben einzufügen, ist die Abbildung der jeweiligen Basis für die dargestellten Automodelle innerhalb der Labelbeschriftung (sofern die Labeltexte im Diagramm integriert sind). Dies kann über das Schlüsselwort `POWERCHARTFORMAT` sowie die Anweisung `ABSINLABELX/Y` erreicht werden und wird dem `POWERCHART`-Befehl vorangestellt.

```

POWERCHARTFORMAT = ABSINLABELX;
PPNUMFORMAT = ' (N=#)';
POWERCHART
TEMPLATE 2
KEY Tab_003
GRATAB GRAPH_01
INVERSE
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT (VARTITLE s)
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION = | ROWS 1:5 COLUMNS 1:5
;
    
```

In diesem Fall heißt die vorangestellte Anweisung `POWERCHARTFORMAT = ABSINLABELX;`, weil die Basenangabe in die Labeltexte des Tabellenkopfes (X-Achse) eingefügt werden sollen. Mit der `PPNUMFORMAT`-Anweisung stellt man außerdem die Darstellung der Zahlenwerte ein, hier mit einem Leerzeichen nach dem Labeltext, einem vorgestellten 'N=' und in runden Klammern.

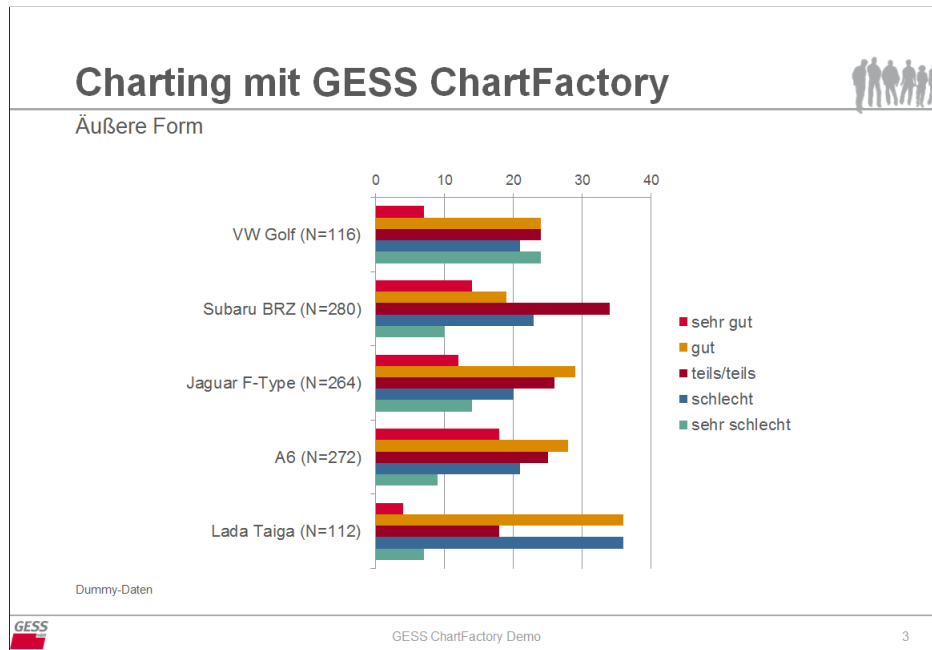


Abb. 3.5: In Labeltexte integrierte Basen-Angaben

Mit `POWERCHARTFORMAT = -ABSINLABELX;` kann die Formateinstellung anschließend wieder zurückgesetzt werden, d.h. auf den folgenden Charts werden ohne neue `POWERCHARTFORMAT`-Anweisung keine Basenangaben eingefügt⁴.

Was ist bisher geschehen?

Auf Grundlage einer einzigen Datentabelle und eines einzigen Vorlagecharts wurden drei PowerPoint-Charts erstellt und dafür drei Kennungen (KEY) verwendet, nämlich Tab_001 bis Tab_003. Dass die Charts immer auf Grundlage desselben Vorlagecharts erstellt wurden, erkennt man daran, dass in jedem der drei Befehle dasselbe Template aufgerufen wurde: `TEMPLATE 2`.

3.4 Beispiel 4: Template p

Während Anweisungen zu Textelementen (`FILL TEXT_[xx] WITH CONCAT`) immer im aktuellen `POWERCHART`-Befehl mitgeführt werden können, gibt es auch die Möglichkeit, mehrere Aktionen für sog. Hauptelemente, also `GRAPH`- und `TABLE`-Elemente, auf einem Chart vorzunehmen. Dafür wird der Aufruf `TEMPLATE p` (`p = previous`) verwendet. So wird gekennzeichnet, dass sich eine Anweisung ergänzend auf das zuletzt bearbeitete Chart bezieht. Zu diesem Zwecke erstellt man zunächst eine neue Vorlage (Seite 3), der — zusätzlich zum Diagramm `GRAPH_01` — zwei Tabellen, die `TABLE_01` und `TABLE_02` genannt werden, und zwei weitere Textfelder hinzugefügt werden.

⁴Voraussetzung: Die Folgecharts werden auf Grundlage einer neuen GESSStabs-Tabelle erstellt.

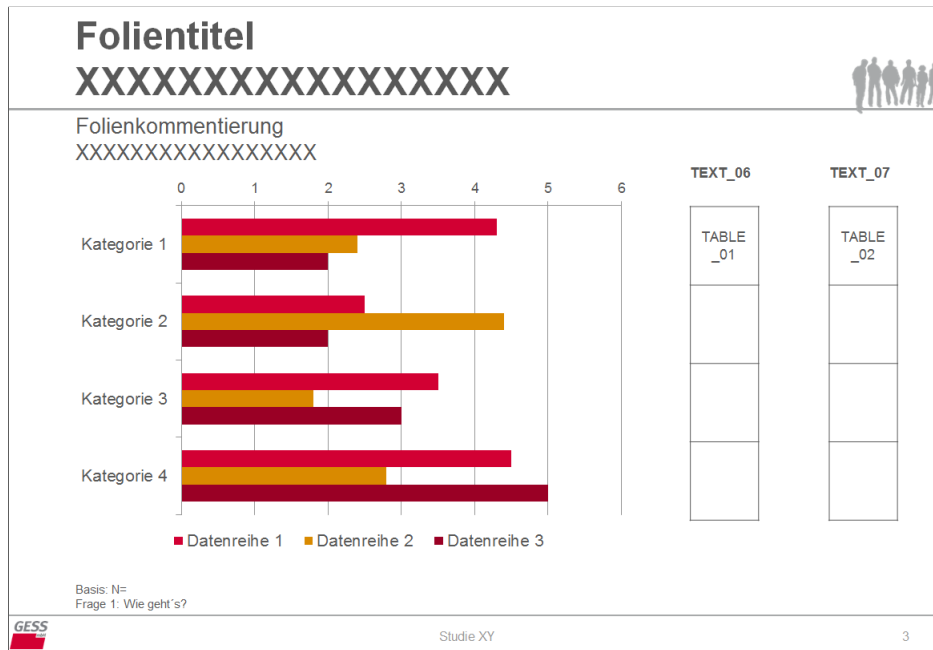


Abb. 3.6: Vorlagechart mit mehreren Hauptelementen

In den rechts stehenden Tabellen sollen nun die Overcodes 'Top 2-Box' und 'Low 2-Box' aus der Tabelle 3.1 dargestellt werden.

Um diese zwei (weiteren) Elemente mit Daten zu befüllen, sind zwei zusätzliche POWERCHART -Anweisungen erforderlich, die sich auf dasselbe Vorlagechart beziehen (TEMPLATE p) wie der Befehl, mit dem das Diagramm befüllt wurde (siehe: Beispiel 2: INVERSE).

Das vollständige Statement sieht so aus:

```

POWERCHART
TEMPLATE 3
KEY Tab_005
GRATAB GRAPH_01
INVERSE
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT (VARTITLE s)
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION = | ROWS 1:5 COLUMNS 1:5
;

POWERCHART
TEMPLATE p
KEY Tab_006
GRATAB TABLE_01
FIXEDTABDIM
INVERSE
NOLABEL X NOLABEL Y
FILL TEXT_06 WITH CONCAT (VALUELABELS Y 65002)
POSITION = | ROWS 6 COLUMNS 1:5
;
    
```

```

POWERCHART
TEMPLATE p
KEY Tab_007
GRATAB TABLE_02
FIXEDTABDIM
INVERSE
NOLABEL X NOLABEL Y
FILL TEXT_07 WITH CONCAT (VALUELABELS Y 65003)
POSITION = | ROWS 7 COLUMNS 1:5
;

```

Die, mittlerweile recht umfangreiche, Syntax hat einige Ergänzungen erfahren:

Der erste Teil bezieht sich auf die Inhalte, die bereits auf den ersten drei Charts abgebildet sind. Einziger Unterschied ist, dass sich nun auf das `TEMPLATE 3` bezogen wird. In den beiden folgenden Befehlen wird diese Vorlage erneut aufgerufen und mit `TEMPLATE p` angezeigt, dass das eben generierte Chart weiter bearbeitet werden soll. In einem nächsten Schritt wird zunächst die links stehende Tabelle (`GRATAB TABLE_01`) bearbeitet. Mit dem Schlüsselwort `FIXEDTABDIM` wird sichergestellt, dass die äußeren Tabellenmaße, die in der Vorlage an die Maße des Diagramms angepasst wurden, unverändert bleiben (unabhängig davon, mit wie vielen Werten die Tabelle gefüllt wird). `INVERSE` wird auch hier eingefügt, um die (in der GESStabs-Tabelle als Zeile aufgeführten) Werte vertikal darzustellen. Mit `NOLABEL X NOLABEL Y` schaltet man die Darstellung der entsprechenden Labeltexte in X- und Y-Richtung aus⁵. `FILL TEXT_06 WITH CONCAT (VALUELABELS Y 65002)` – dieser Befehlsteil bedeutet: Fülle die Textbox namens `TEXT_06` mit dem Labeltext (`VALUELABELS`) des in der Datentabelle in der Zeile (`Y`) dargestellten Labels mit dem Code 65002 (Top-2-Box). Als darzustellenden Tabellenbereich wird mit `POSITION = | ROWS 6 COLUMNS 1:5` die sechste Zeile und alle fünf Spalten aus der Datentabelle bestimmt. Analog läuft es für die rechts stehende Tabelle (`TABLE_02`) auf dem Chart.

⁵Bei der Befüllung von Diagrammen und Tabellen übernimmt die Software automatisch die Labeltexte der ausgewählten Datenreihen. Würde dies an dieser Stelle nicht deaktiviert werden, würde GESS ChartFactory also direkt in den Tabellen eine weitere Spalte sowie Zeile mit den Variablenlabels einfügen. Dies ist nicht erwünscht, da sich in den PowerPoint-Tabellen auf die Beschriftungen im Diagramm bezogen wird.

Daraus wird folgendes Chart generiert:

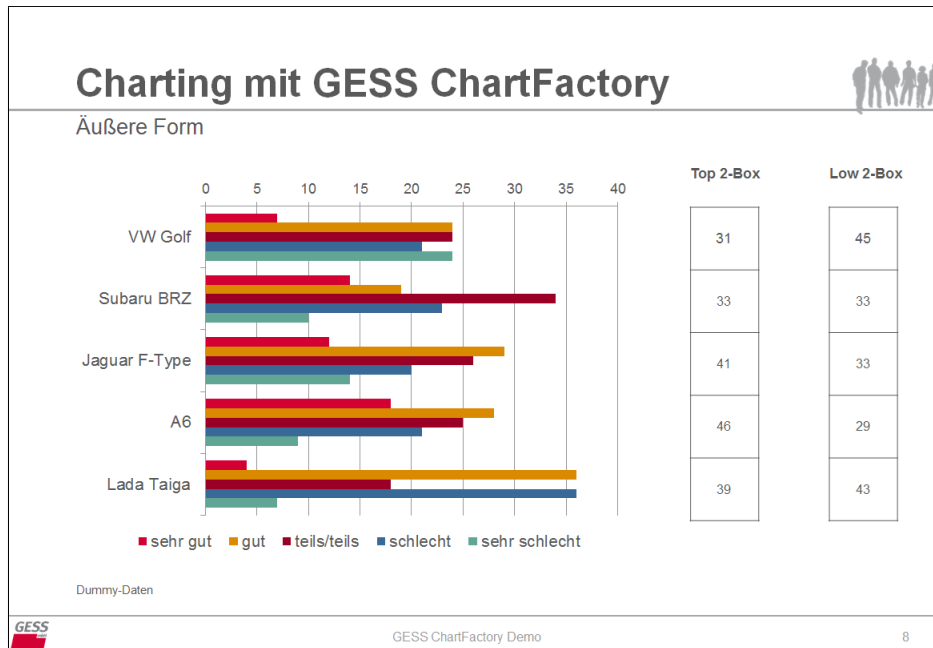


Abb. 3.7: Ausgabechart mit mehreren Hauptelementen

3.5 Beispiel 5: einfaches Liniendiagramm

Nun werden Mittelwerte aufgegriffen, die die Bewertung der Zufriedenheit mit den bekannten Automodellen von sechs verschiedenen Gruppen abbilden:

Tab. 3.2: Mittelwerte-Tabelle

GESS ChartFactory Demo					
Abs.	Modellvariante				
	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
Base	116	280	264	272	112
Gruppe 1	3.21	3.01	2.91	3.07	2.68
Gruppe 2	2.72	3.19	3.11	3.19	2.50
Gruppe 3	3.41	2.84	3.18	3.24	2.93
Gruppe 4	3.52	3.20	2.65	3.03	3.00
Gruppe 5	2.93	3.04	2.86	2.78	2.89
Gruppe 6	2.38	3.24	2.98	2.84	2.82

GESS mbH

Diese sollen in einem einfachen Liniendiagramm dargestellt werden, das auf einem neuen Vorlagechart (Nr. 4) GRAPH_01 genannt wird:

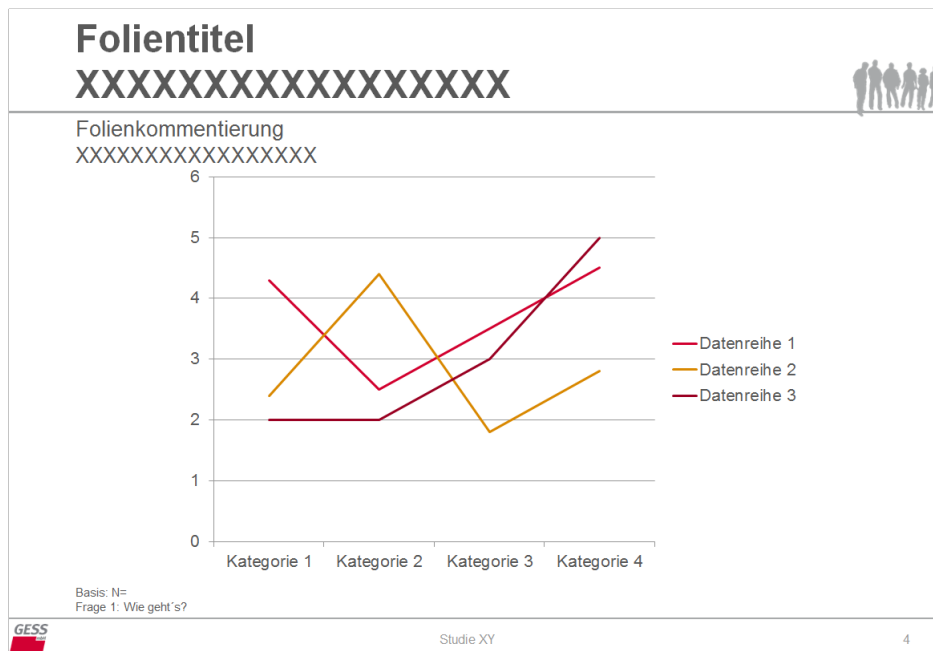


Abb. 3.8: Vorlage für ein einfaches Liniendiagramm

Das TABLE-Statement dazu lautet: `OVERVIEW = k BY MEAN (z1 z2 z3 z4 z5 z6);`

Der POWERCHART-Befehl sieht so aus:

```
POWERCHART
TEMPLATE 4
KEY Tab_008
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT ("Zufriedenheit nach Gruppe")
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION =
| ROWS 1:6 COLUMNS 1:5
;
```

Mit folgendem Resultat:

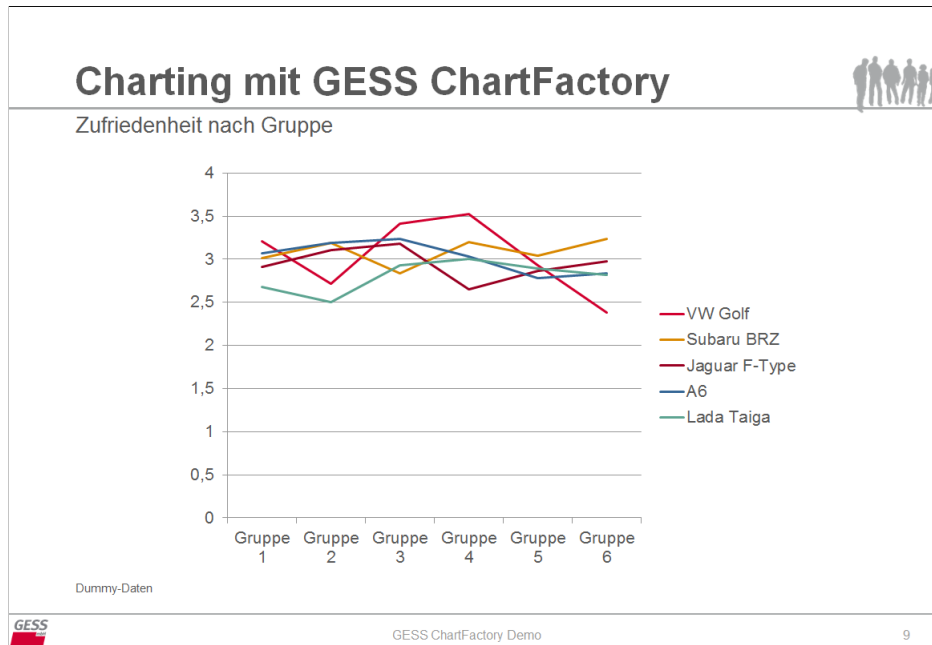


Abb. 3.9: Ausgabe für ein einfaches Liniendiagramm

Wie man sieht, hat sich auch hier das Diagramm aus der Vorlage an die Daten angepasst: Datenreihen wurden hinzugefügt, die Skala hat den passenden Wertebereich erhalten und die Achsen wurden entsprechend befüllt und beschriftet.

3.6 Beispiel 6: START

Bei Betrachtung dieses Charts könnte die Zufriedenheit mit dem VW Golf aufgrund der starken Schwankungen über die Gruppen besonders interessieren — mithilfe eines Kategoriencharts ist eine genauere Betrachtung möglich.

Dazu erstellt man folgende Übersichts-Tabelle (`TABLE = k by z1 z2 z3 z4 z5 z6;`), aus der uns eigentlich nur die erste Spalte interessiert:

Tab. 3.3: Bewertung der Automarken nach Kategorien

GESS ChartFactory Demo					
Col % Basis	Modellvariante				
	VW Golf 116	Subaru BRZ 280	Jaguar F-Type 264	A6 272	Lada Taiga 112
Gruppe 1					
sehr zufrieden	17 %	11 %	14 %	9 %	21 %
zufrieden	14 %	33 %	32 %	28 %	29 %
teils/teils	21 %	17 %	21 %	25 %	18 %
unzufrieden	28 %	20 %	17 %	21 %	25 %
sehr unzufrieden	21 %	18 %	17 %	16 %	7 %
sehr gut, gut	31 %	44 %	45 %	37 %	59 %
schlecht, sehr schlecht	48 %	39 %	33 %	37 %	32 %
Gruppe 2					
sehr zufrieden	10 %	9 %	6 %	9 %	14 %
zufrieden	38 %	23 %	23 %	22 %	39 %
teils/teils	31 %	24 %	36 %	24 %	29 %
unzufrieden	10 %	30 %	24 %	32 %	18 %
sehr unzufrieden	10 %	14 %	11 %	13 %	0 %
sehr gut, gut	48 %	31 %	29 %	31 %	54 %
schlecht, sehr schlecht	21 %	44 %	35 %	46 %	18 %
Gruppe 3					
sehr zufrieden	7 %	14 %	11 %	9 %	7 %
zufrieden	17 %	27 %	21 %	19 %	39 %
teils/teils	21 %	24 %	20 %	29 %	25 %
unzufrieden	38 %	29 %	36 %	25 %	11 %
sehr unzufrieden	17 %	6 %	12 %	18 %	18 %
sehr gut, gut	24 %	41 %	32 %	28 %	46 %
schlecht, sehr schlecht	55 %	34 %	48 %	43 %	29 %
Gruppe 4					
sehr zufrieden	10 %	9 %	20 %	13 %	14 %
zufrieden	10 %	24 %	24 %	25 %	25 %
teils/teils	21 %	23 %	32 %	19 %	25 %
unzufrieden	34 %	27 %	20 %	31 %	18 %
sehr unzufrieden	24 %	17 %	5 %	12 %	18 %
sehr gut, gut	21 %	33 %	44 %	28 %	39 %
schlecht, sehr schlecht	59 %	44 %	24 %	43 %	36 %
Gruppe 5					
sehr zufrieden	14 %	7 %	18 %	10 %	18 %
zufrieden	31 %	26 %	28 %	35 %	25 %
teils/teils	21 %	35 %	17 %	29 %	21 %
unzufrieden	17 %	19 %	23 %	16 %	21 %
sehr unzufrieden	17 %	13 %	14 %	9 %	14 %
sehr gut, gut	45 %	33 %	46 %	46 %	43 %
schlecht, sehr schlecht	34 %	32 %	37 %	25 %	36 %
Gruppe 6					
sehr zufrieden	31 %	7 %	15 %	21 %	29 %
zufrieden	21 %	21 %	20 %	16 %	18 %
teils/teils	31 %	26 %	26 %	29 %	18 %
unzufrieden	14 %	31 %	30 %	26 %	14 %
sehr unzufrieden	3 %	14 %	9 %	7 %	21 %
sehr gut, gut	52 %	29 %	35 %	37 %	46 %
schlecht, sehr schlecht	17 %	46 %	39 %	34 %	36 %

Und folgende Vorlage:

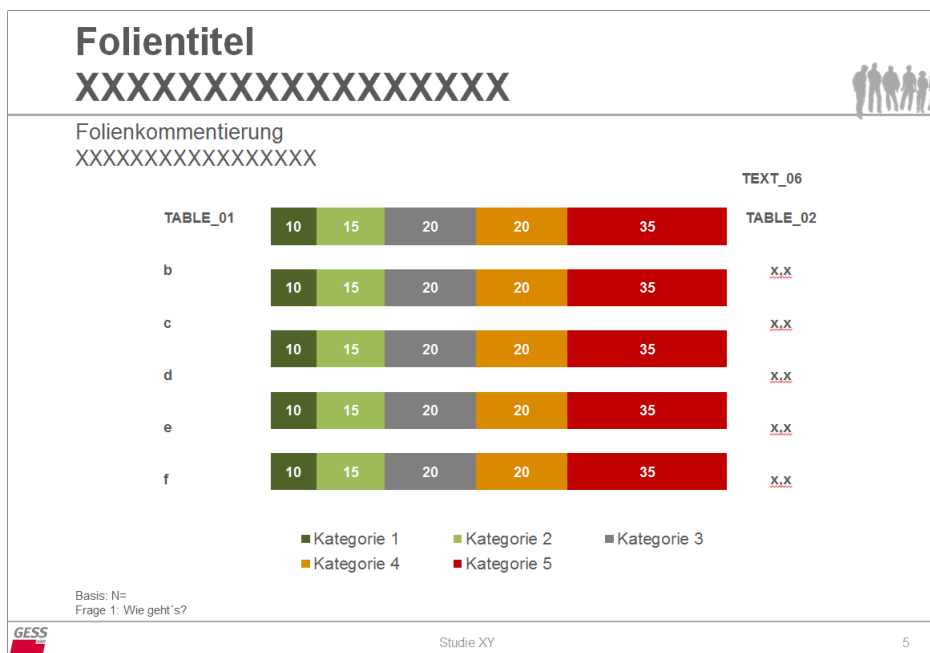


Abb. 3.10: Vorlage für ein Kategoriendiagramm

Zunächst kümmert man sich um das Kategoriendiagramm. In die waagrecht verlaufenden Balken sollen jeweils die fünf Kategorien aus der ersten Tabellenspalte (VW Golf) für die sechs Gruppen gefüllt werden. Diese kann man *nicht* einfach mit `POSITION = | ROWS 1:30 COLUMNS 1;` realisieren.

Der Grund liegt in der Struktur des Diagramm-Datenblattes (im PowerPoint-Menü unter Entwurf → 'Daten bearbeiten' zu finden):

	A	B	C	D	E	F	G
1		Datenreihe 1	Datenreihe 2	Datenreihe 3	Datenreihe 4	Datenreihe 5	
2	Kategorie 1	10	10	10	10	10	
3	Kategorie 2	15	15	15	15	15	
4	Kategorie 3	20	20	20	20	20	
5	Kategorie 4	20	20	20	20	20	
6	Kategorie 5	35	35	35	35	35	
7							

Abb. 3.11: Datenblatt des Kategoriendiagrammes

Riefe man den Tabellenbereich wie eben genannt auf, würden die Daten aus Tabellenspalte 1 im Datenblatt in Spalte B senkrecht heruntergeschrieben und nicht korrekt im Diagramm abgebildet werden⁶.

Stattdessen muss die erste Spalte aus der Tabelle in die sechs Balkenbereiche gesplittet und den entsprechenden Bereichen im Datenblatt zugeordnet werden. Dazu sind sechs POWERCHART-Befehle notwendig. Der erste Befehl lautet:

```
POWERCHART
KEY Tab_009
TEMPLATE 5
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT ("Zufriedenheit mit VW Golf")
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
NOLABEL X
START A2
POSITION = | ROWS 1:5 COLUMNS 1
;
```

Mit dem POSITION-Operator werden die erste Spalte und die ersten fünf Zeilen in unserer GESStabs-Datentabelle ausgewählt und mit NOLABEL X die Anzeige der Labeltexte (bzw. des Labeltextes, hier: VW Golf) ausgeschaltet. START A2 steuert, an welcher Stelle im PowerPoint-Datenblatt mit dem Befüllen der Daten begonnen wird und zwar in Zelle A2: Die ausgewählten fünf Zahlen werden dann senkrecht in Spalte B geschrieben, die Labeltexte werden der Datenreihe in Spalte A vorangestellt.

```
POWERCHART
KEY Tab_010
TEMPLATE p
GRATAB GRAPH_01
NOLABEL X NOLABEL Y
START C2
POSITION = | ROWS 6:10 COLUMNS 1
;
```

Im zweiten POWERCHART-Befehl wird dieselbe Vorlage und dasselbe Diagramm aufgerufen und mit den Kategorien der nächsten Variable (Zeilen 6 bis 10) befüllt. Diese sollen im zweiten Balken abgebildet werden, die Dateneingabe beginnt demnach in Zelle C2 des Datenblattes. Weil

⁶Pragmatisch wäre, einen Schritt zurückzugehen und die GESStabs-Tabelle an die Beschaffenheit der Chartvorlage anzupassen. In diesem Fall wird die bestehende GESStabs-Tabelle aber als gegeben betrachtet.

mit dem ersten Befehl schon einmal die Kategorie-Namen niedergeschrieben wurden, ist dies nun nicht mehr gewünscht: `NOLABEL Y`.

Analog werden die restlichen Balken mit Daten aus den entsprechenden Tabellenbereichen gefüllt⁷.

Nun sollen auf demselben Chart noch eine Datenbeschriftung (links) für das Balkendiagramm⁸ und eine Mittelwerte-Übersicht (rechts) erscheinen. Dafür greift man auf die vorherige Mittelwertetabelle zurück und schreibt die folgenden `POWERCHART`-Befehle: Zuerst wird die für die Labeltexte vorgesehene Tabelle (`TABLE_01`) und dann die Tabelle für die Mittelwerte (`TABLE_02`) des Vorlage-Charts aufgerufen und mit Informationen befüllt.

```
POWERCHART
KEY Tab_016
TEMPLATE p
GRATAB TABLE_01
FIXEDTABDIM
POSITION = | ROWS 1:6 COLUMNS LABELS
;

POWERCHART
KEY Tab_017
TEMPLATE p
GRATAB TABLE_02
FILL TEXT_06 WITH CONCAT ("Mittelwerte")
FIXEDTABDIM
NOLABEL X NOLABEL Y
POSITION = | ROWS 1:6 COLUMNS 1
;
```

Neu ist hier `COLUMN LABELS` in der `POSITION`-Angabe des ersten Befehls: Da die Variablenlabels nicht direkt Teil der Datentabelle in `GESStabs` sind und so nicht über die Angabe ihrer Position aufgerufen werden können, kann stattdessen das Schlagwort `LABELS` benutzt werden, das in diesem Falle bewirkt: Schreibe die Labeltexte der ersten sechs Reihen in die `TABLE_01`⁹

⁷Für mehrere gleichartige Befehle bietet sich der Einsatz eines Makros an, siehe [Einbindung von Makros](#).

⁸Die Beschriftung integriert man in diesem Fall nicht direkt in das Diagramm, weil dort als Variablenname dann fünfmal 'VW Golf' stünde. Es soll aber mit 'Gruppe 1 bis 6' beschriftet werden, daher wird ein anderer Weg gewählt.

⁹In diesem Fall handelt es sich, genau genommen, gar nicht um Labeltexte, sondern um die Variablennamen der Variablen `z1` bis `z6`. Diese werden aber in der `GESStabs`-Tabelle wie eben solche behandelt, sodass sie mit dem `LABELS`-Schlagwort greifbar sind (siehe [Berechnete Tabellenelemente](#)).

Das ist das fertige Chart:

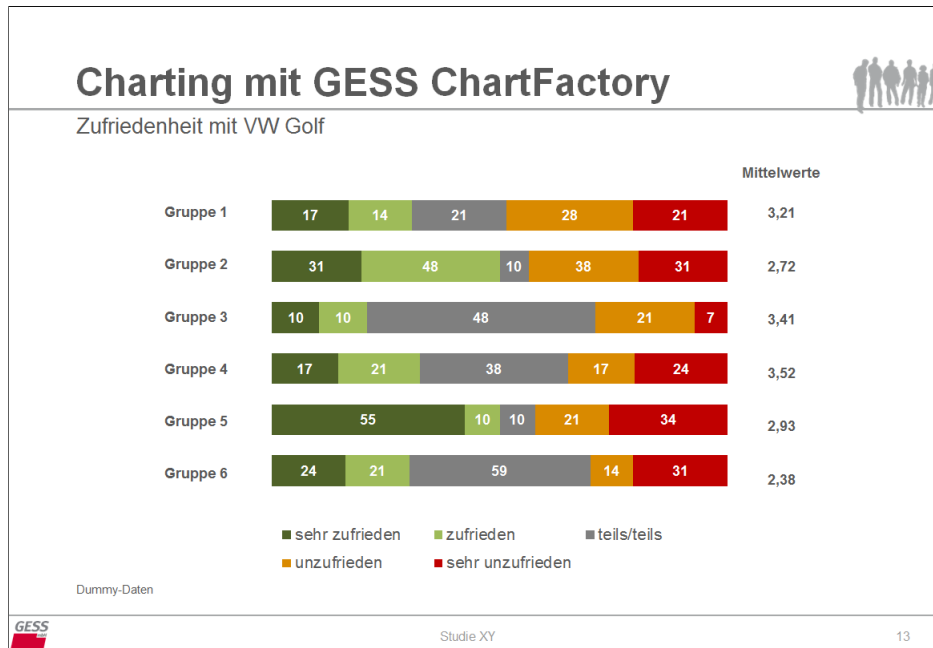


Abb. 3.12: Ausgabe mit Kategoriendiagramm

3.7 Beispiel 7: DIRECTION FALLING

An einem anderen Diagrammtyp wird deutlich, was auch bei der Befüllung des Kategoriencharts angewandt wurde: Bei der Verbindung von GESSStabs-Datentabellen und PowerPoint-Charts geht es häufig darum, erstere im Zuge der Übertragung an die Beschaffenheit der PowerPoint-Elemente anzupassen.

Angenommen, man möchte auf Grundlage der folgenden Datentabelle die Verteilung der sehr zufriedenen (Code 1) und der sehr unzufriedenen (Code 5) Befragten aus Gruppe 4 über alle Automodelle vergleichen:

GESS ChartFactory Demo							
Col %	Base	Total	Gruppe 4				
			sehr zufrieden	zufrieden	teils/teils	unzufrieden	sehr unzufrieden
Base	1044		140	240	252	272	140
Modellvariante							
VW Golf	116	11 %	9 %	5 %	10 %	15 %	20 %
Subaru BRZ	280	27 %	17 %	28 %	25 %	28 %	34 %
Jaguar F-Type	264	25 %	37 %	27 %	33 %	19 %	9 %
A6	272	26 %	26 %	28 %	21 %	31 %	23 %
Lada Taiga	112	11 %	11 %	12 %	11 %	7 %	14 %

GESS mbH

Tab. 3.4: Zufriedenheit mit Automarken, Gruppe 4

Dafür verwendet man dieses Vorlage-Chart, in das links die Werte der sehr unzufriedenen und rechts die der sehr zufriedenen Befragten eintragen werden sollen.

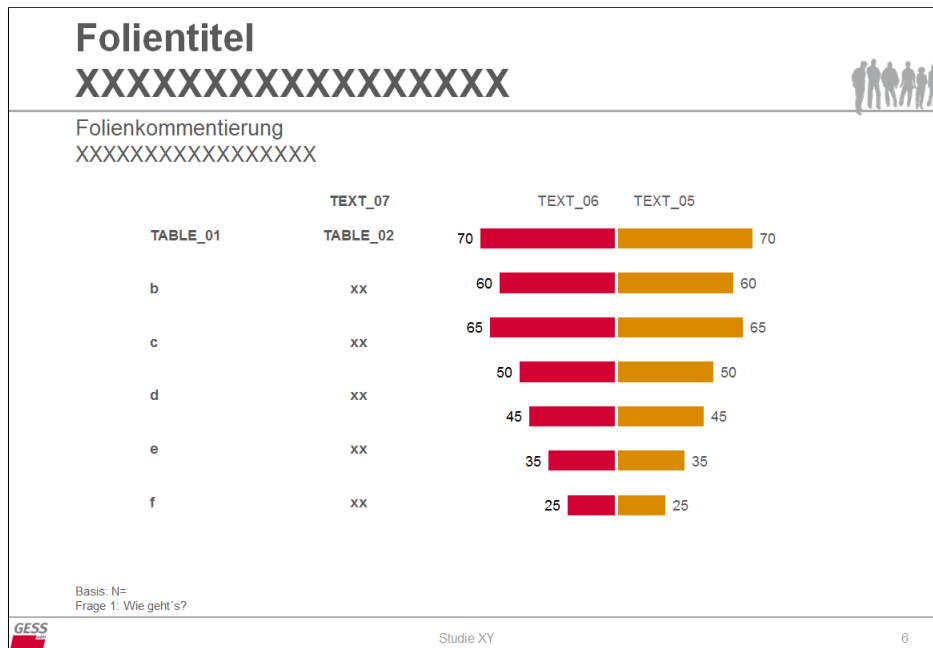


Abb. 3.13: Chartvorlage mit positiv und negativ gerichteten Werten

Schaut man sich die Datentabelle des Diagramms an, sieht man, dass die nach links verlaufenden Balken (Datenreihe 1) negative Vorzeichen aufweisen.

	A	B	C	D
1		Datenreihe 1	Datenreihe 12	
2	Kategorie 1	-70	70	
3	Kategorie 2	-60	60	
4	Kategorie 3	-65	65	
5	Kategorie 4	-50	50	
6	Kategorie 5	-45	45	
7	Kategorie 6	-35	35	
8	Kategorie 7	-25	25	
9	Ziehen Sie zum Ändern der Größe des Diagramm			

Abb. 3.14: Datentabelle mit positiv und negativ gerichteten Werten

Die Befüllung dieser Daten kann mithilfe des `DIRECTION FALLING`-Statements umgesetzt werden. Weil innerhalb eines `POWERCHART`-Befehls nicht 'die Richtung' gewechselt werden kann, schreibt man für die linke und rechte Diagramm-Seite zwei getrennte `POWERCHART`-Anweisungen. Der Datenbereich, der 'nach links zeigen' soll, wird mit einem entsprechenden Statement gelenkt.

Außerdem kommen zwei `POWERCHART`-Befehle zum Füllen der Tabellen hinzu. Die Syntax für das vollständige Chart sieht so aus:

```
POWERCHART
KEY Tab_018
TEMPLATE 6
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT ("Sehr zufriedene und sehr unzufriedene Befragte (
Gruppe 4)")
```

```

FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
FILL TEXT_06 WITH CONCAT (VALUELABELS X 5)
DIRECTION FALLING
FIXEDRANGE
POSITION = | ROWS 1:5 COLUMNS 5
;

POWERCHART
KEY Tab_019
TEMPLATE p
GRATAB GRAPH_01
FILL TEXT_05 WITH CONCAT (VALUELABELS X 1)
NOLABEL Y NOLABEL X
START C2
POSITION = | ROWS 1:5 COLUMNS 1
;

POWERCHART
KEY TAB_020
TEMPLATE p
GRATAB TABLE_01
FIXEDTABDIM
POSITION = | ROWS 1:5 COLUMNS LABELS
;

POWERCHART
KEY Tab_021
TEMPLATE p
GRATAB TABLE_02
FILL TEXT_07 WITH CONCAT ("Basen")
FIXEDTABDIM
POSITION = | ROWS 1:5 COLUMNS ABCOLUMN
;

```

Die erste Anweisung bezieht sich auf die linke Diagrammseite. In diesem Zuge werden allerhand Textelemente auf dem Chart befüllt. Interessant ist hier vor allem die automatisierte Beschriftung der linken Diagrammseite via `FILL TEXT_06 WITH CONCAT (VALUELABELS X 5)` mit dem Text des darzustellenden Labels (X-Variable und Labelcode 5 = sehr unzufrieden). Dann wird das Diagramm aufgegriffen (`GRATAB GRAPH_01`). In diesem sollen die Daten in 'umgekehrter Richtung' dargestellt werden (`DIRECTION FALLING`). Mit `FIXEDRANGE` wird sichergestellt, dass die Formateinstellungen der Diagrammvorlage (z.B. Balkenfarben) bestehen bleiben. Als Tabellenbereich werden alle Zeilen und die letzte Spalte der GESStabs-Tabelle ausgewählt.

Der zweite Befehl greift auf dasselbe Diagramm, jedoch auf dessen rechte Hälfte, zu und ist ganz ähnlich: Hier möchten man die Daten in 'normaler Richtung' anzeigen und lässt daher die `DIRECTION FALLING`-Spezifikation weg¹⁰. Begonnen wird in Zelle C2 (`START C2`) des Datenblattes mit der Datenbefüllung, die Datenbeschriftungen wird diesmal ausgelassen (`NOLABEL X NOLABEL Y`). Die erste Spalte der GESStabs-Tabelle wird ausgewählt und deren Labeltext im darüberstehenden Textfeld eingefügt.

¹⁰`DIRECTION RISING` würde dasselbe bewirken.

Die beiden Befehle zur Befüllung der Tabellen erklären sich mittlerweile von selbst, neu ist hier nur die Spezifikation im `POSITION`-Statement des letzten `POWERCHART`-Befehls, in der direkt die Basen-Spalte aus der Tabelle angesteuert wird (`COLUMNS ABSCOLUMN`).

Und das ist das fertige Chart:

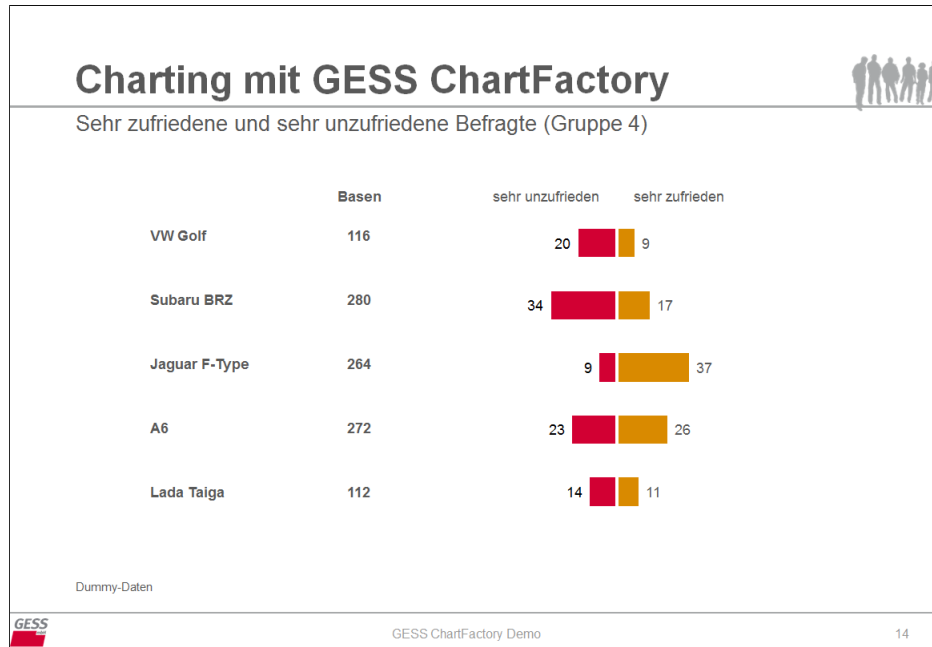


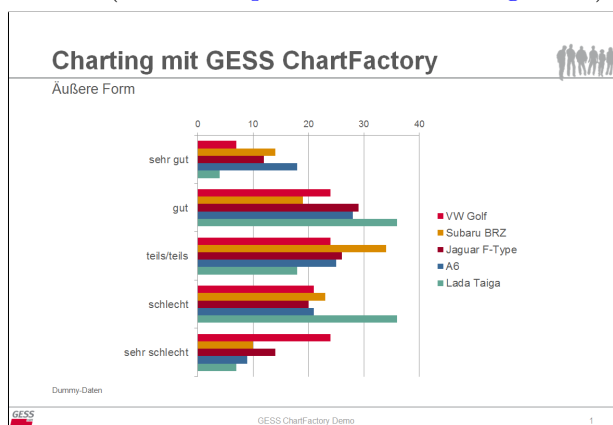
Abb. 3.15: Ausgabechart mit 'DIRECTION FALLING'

4 Texte

Texte können mit GESS ChartFactory auf drei unterschiedliche Wege auf einem PowerPoint-Chart platziert werden.

4.1 In Diagrammen

Erstens können Labeltexte (oder andere Bezeichner-Texte) direkt ins Diagramm eingespielt werden (siehe: [Beispiel 1: einfaches Diagramm](#)).



Dazu benötigt man lediglich ein entsprechend vorbereitetes Vorlagediagramm. Die Darstellung der Labeltexte erfolgt dann automatisch für die ausgewählten Wertebereiche der Datentabelle. Deaktiviert werden kann dies, indem man die Übertragung der Labeltexte in das Diagramm via `NOLABEL X / Y` unterdrückt.

Abb. 4.1: Diagramm mit integrierten Labeltexten

Vorteile dieser Option sind einerseits die einfache und schnelle (weil automatische) Umsetzung und andererseits weitere mögliche Ergänzungen, die durch das Einfügen einfacher Schlüsselwörter aktiviert werden können. Dazu zählt etwa das automatische Anfügen von Basenangaben an die Labeltexte mittels `POWERCHARTFORMAT`-Statement (siehe 3.3).

Ein Nachteil ist, dass die Texte nicht einzeln formatiert werden können — die Formatierungsoptionen beschränken sich hier auf die Möglichkeiten, die es in PowerPoint für in Diagramme integrierte Texte gibt, und können beispielsweise nur einheitlich für alle Texte einer Diagramm-Achse eingestellt werden¹.

4.2 Textboxen

Möglich ist die Formatierung einzelner Texte aber, wenn diese in Textboxen² stehen. Für das oben stehende Chart wird dies für die Labeltexte umgesetzt, indem die Achsenbeschriftung innerhalb des Diagrammes auf dem Vorlagechart ausgeblendet und vor dem Diagramm einzelne Textboxen platziert werden (`TEXT_05` bis `TEXT_09`) — dies ist die zweite Möglichkeit zur Übergabe von Texten an PowerPoint.

¹Siehe hierzu auch Kapitel 10.2.

²oder in Tabellen

Entsprechend umgestaltet sieht unsere Vorlage so aus:

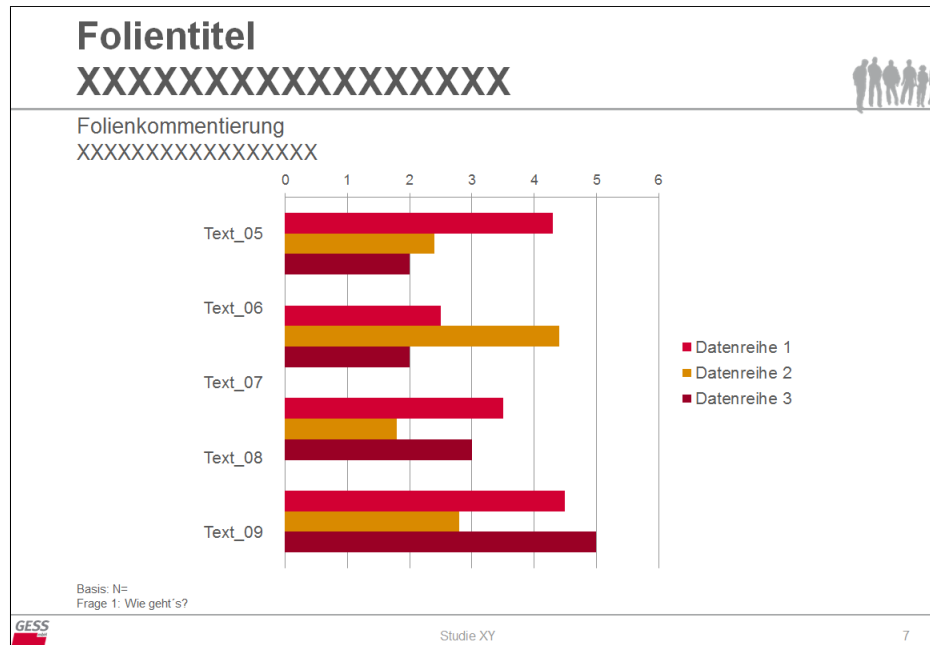


Abb. 4.2: Vorlage mit Textboxen

Die Textelemente werden mithilfe des Statements `FILL TEXT_[xx] WITH CONCAT` angesprochen und mit freien Texten und/oder GESStabs-Elementen befüllt. Hier sollen die einzelnen Textboxen mit den Labeltexten der auf der X-Achse dargestellten Variablen befüllt werden, also: `FILL TEXT_05 WITH CONCAT (VALUELABELS X 1)`, `FILL TEXT_06 WITH CONCAT (VALUELABELS X 2)` usw...

In der Fußnote einer PowerPoint-Folie werden zudem häufig Informationen wie Fragetext, dargestellte statistische Maßzahl und zugrundeliegende Basis der dargestellten Inhalte vermerkt. In diesem Fall sollen ein freier Text und die (gewichtete) Fallzahl abgebildet werden. Diese beiden (und grundsätzlich beliebig viele) Elemente können verkettet und als Fließtext in die unten stehende Textbox eingefügt werden: `FILL TEXT_05 WITH CONCAT ("Dummy-Daten, Basis = " ABSOLUTE)` – auf das Schlagwort `CONCAT` folgen in runden Klammern der freie Text 'Dummy-Daten, Basis = ' in Anführungszeichen sowie das GESStabs-Element `ABSOLUTE`.

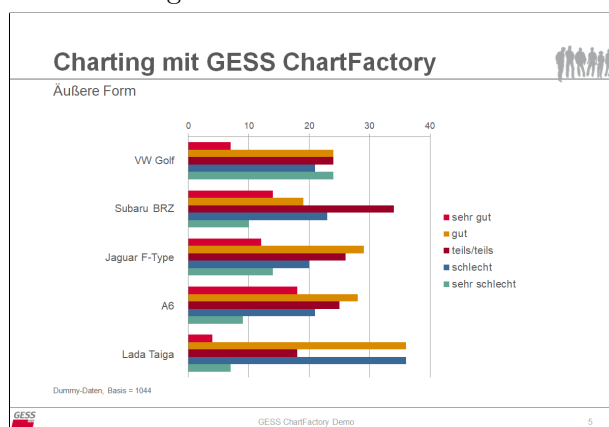


Abb. 4.3: Labeltexte in Textboxen

Das Ergebnis ist links zu sehen und unterscheidet sich optisch nicht vom ersten Chart (abgesehen vom leicht angepassten Fußnoten-Text). Nun kann man aber (im Gegensatz zur ersten Lösung) an die einzelnen Labeltexte herangehen und diese in der PowerPoint-Vorlage formatieren oder individuell positionieren.

Textboxen lassen sich nun durch Einschub eines FONT-Statements beliebig und direkt im POWERCHART-Statement formatieren. Möchte man etwa die Beschriftung des VW Golf besonders betonen, können im Rahmen der FILL TEXT_[xx] WITH FREETEXT-Option Anweisungen zur Schriftart, -farbe und -größe integriert werden:

```
FILL TEXT_05 with FREETEXT "<FONT BOLD ITALIC COLOR=0,0,255 SIZE=+4>VW Golf</FONT>"
```

Mit FONT BOLD ITALIC wird eine fette und kursive Schrift erzeugt, COLOR=0,0,255 bewirkt eine Blaufärbung und SIZE=+4 führt dazu, dass die Schriftgröße gegenüber der voreingestellten Textgröße des Textelements um 4 pt. vergrößert wird. Im Zwischenbereich wird der zu formatierende Freitext beschrieben.

Eine weitere Möglichkeit für den individualisierten Einsatz von Textboxen ist, eine im Template bestehende Textbox durch eine Grafik zu ersetzen. Auch hier kommt die FREETEXT-Option zur Anwendung. Möchte man etwa den Text 'A6' durch ein Audi-Logo ersetzen, gelingt dies so: FILL TEXT_08 WITH FREETEXT WITH "LabelIconReplace=pos=r,resize=B;Pic_Audi". Auf das Schlagwort LabelIconreplace folgt die rechtsbündige Ausrichtung des Icons (pos=r), dessen proportionale Skalierung zur Textboxhöhe und -breite mittels resize=B und die Benennung des einzufügenden Bildes³.

Das Ergebnis beider Anpassungsoptionen sieht so aus:

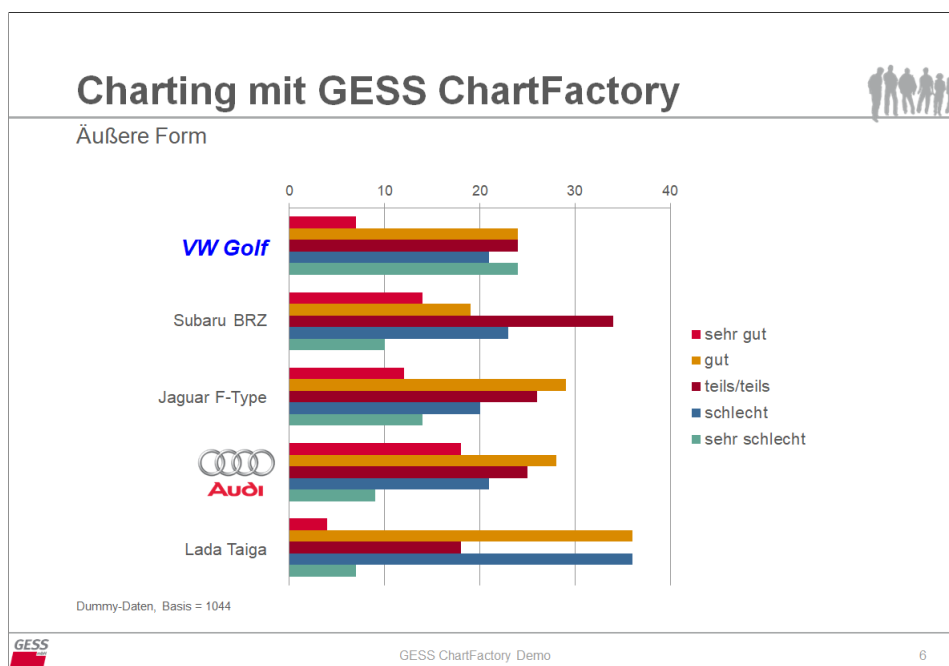


Abb. 4.4: Gestaltungsmöglichkeiten von Textboxen

Beachte: Die Formatierung sowie der Ersatz von Textboxen durch eine Grafik funktionieren nur innerhalb der FREETEXT-Option und *nicht* unter Verwendung von FILL TEXT_[xx] WITH CONCAT bzw. beim Aufruf von GESStabs-Elementen.

Der Vorteil beim Einsatz von Textboxen ist gleichzeitig ihr Nachteil: Die einzelnen Textelemente können gezielt und individuell angesprochen, ausgerichtet und gestaltet werden. Dadurch ist eine flexible und völlig freie Gestaltung möglich. Andererseits erfordert dies relativ viel Arbeit

³Das einzusetzende Icon muss auf der ersten Seite der Template-Datei abgelegt werden. Alle Spezifikationsvarianten der LabelIconReplace-Option werden im Abschnitt zur verwandten POWERCHARTOPTION LabelIconReplace beschrieben.

an der Vorlagedatei, da sich die Textfelder beispielsweise nicht automatisch an die räumliche Verteilung der Balken im Diagramm anpassen. D.h. es muss vorher bekannt sein, wie viele Kategorien dargestellt werden, um entsprechend viele Textboxen einrichten zu können. Dies macht dann eventuell eine weitere Nachbereitung des Vorlagecharts nötig (d.h. Ausprobieren der Chartausgabe und Neupositionierung der einzelnen Elemente), um die Ausrichtung der Textelemente auf dem Chart und im Verhältnis zu anderen Elementen zu optimieren.

4.3 Tabellen

Möchte man sich diese Kleinarbeit sparen, können Texte drittens als **Tabelle** auf dem Chart platziert werden.

Dazu erstellt man auf dem Vorlagechart eine Tabelle anstelle der einzelnen Textelemente und benennt diese mit `TABLE_01`.

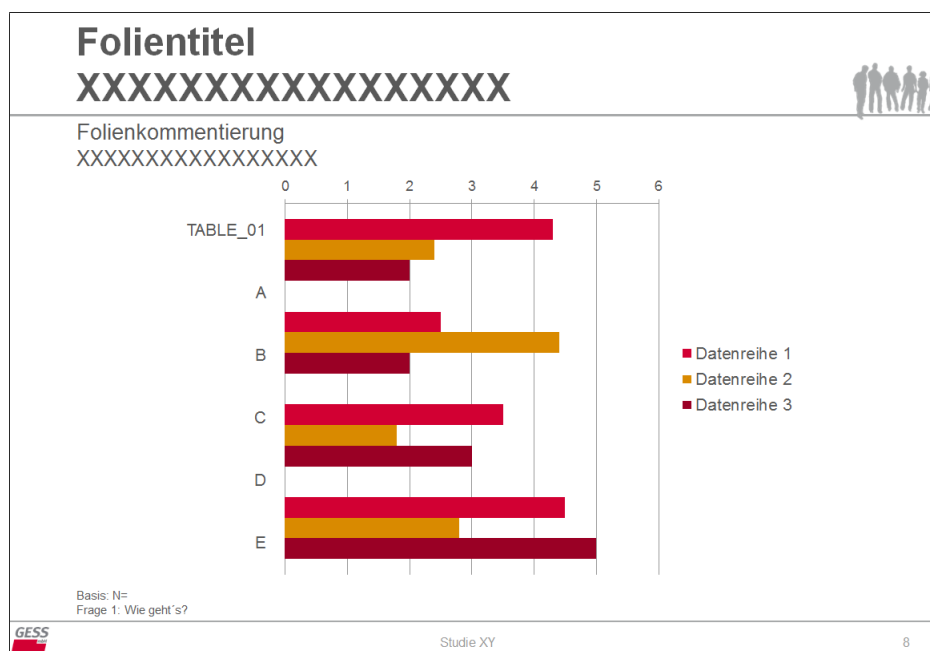


Abb. 4.5: Vorlage mit Texttabelle

Der Unterschied zu den vorherigen Möglichkeiten besteht darin, dass das Befüllen der Tabelle mit Labeltexten eine eigene (vom Statement für das Diagramm abgetrennte) POWERCHART-Anweisung benötigt. Hier die vollständige Syntax:

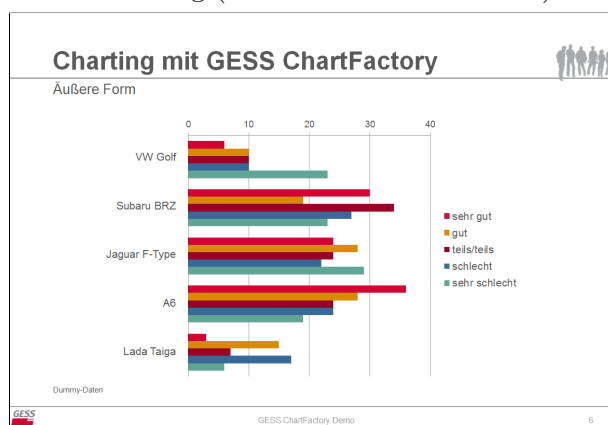
```
TABLE = s BY k;

POWERCHART
TEMPLATE 8
KEY Tab_022
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT (VARTITLE s)
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION = | ROWS 1:5 COLUMNS 1:5
;

POWERCHART
TEMPLATE p
KEY Tab_023
GRATAB TABLE_01
FIXEDTABDIM
POSITION = | ROWS 1:5 COLUMNS LABELS
;
```

Interessant ist hier vor allem der zweite Befehl: Es wird dasselbe Vorlagechart aufgegriffen wie zuvor (TEMPLATE p) und im POSITION-Statement bestimmt, dass die Labeltexte (COLUMNS LABELS) der ersten fünf Labels (ROWS 1:5) in TABLE_01 zu schreiben sind.

Das Ergebnis sieht genauso aus wie auf den ersten beiden Charts, ist aber im Gegensatz zur zweiten Lösung (zumindest in diesem Fall) effektiver:



Wie man sieht, haben sich beide Grafiken auf dem Vorlagechart an die Anzahl der darzustellenden Elemente angepasst. Während das Diagramm automatisch seine äußeren Maße beibehält, reagieren die Außenmaße von Tabellen darauf, wenn weniger Elemente darzustellen sind als in der Vorlage vermerkt: Die Zellen behalten ihren inneren Zeilenabstand bei, die Tabelleaußenmaße werden also kleiner. Mit FIXEDTABDIM kann dem entgegengewirkt und die äußere Dimension der Tabelle beibehalten werden.

Abb. 4.6: Labeltexte in Texttabelle

Ein weiterer Vorteil, Texte in Tabellenform auf Charts einzubinden, ist die MARKSTRING-Option. Hiermit können — als Alternative zum FONT-Statement, das auch für Tabellen funktioniert — einzelne Texte oder -teile direkt über den POWERCHART-Befehl formatiert werden. Möchte man beispielsweise das erste Label fett und rot markieren, fügen wir das Statement `MARKSTRING "VW Golf" MARKBOLD FOREGROUND $FF0000` in den zweiten POWERCHART-Befehl ein. Das bewirkt Folgendes: Die Texte in der Tabelle werden daraufhin untersucht, ob sie die

als MARKSTRING angegebene Zeichenfolge (hier: VW Golf⁴) enthalten. Ist dies der Fall, werden MARKBOLD und FOREGROUND <color> (hier: \$FF0000 für knallrot) ausgeführt.

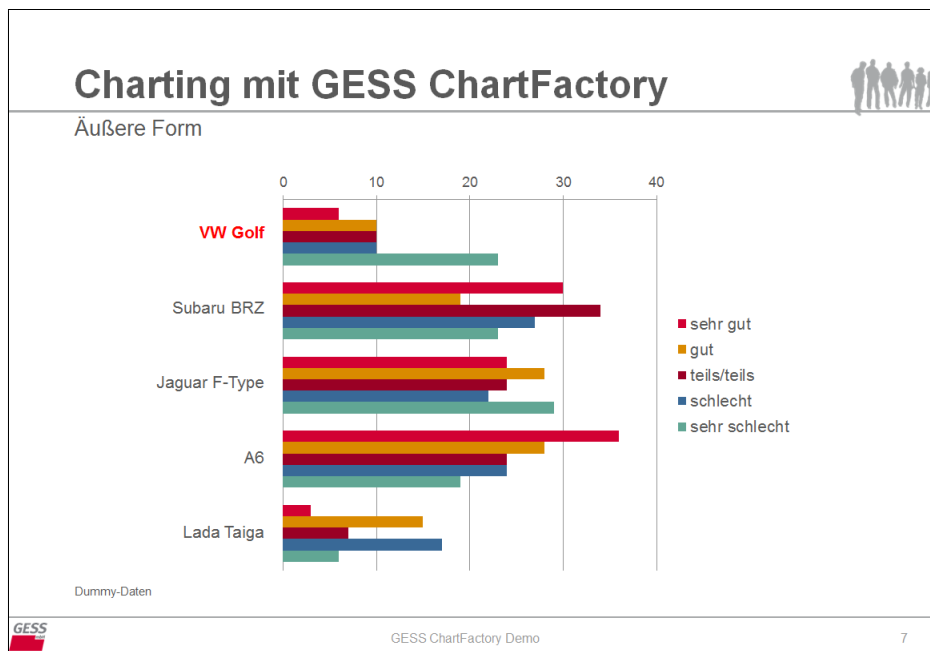


Abb. 4.7: Textformatierung mit 'MARKSTRING'

Der Vorteil gegenüber des FONT-Statements ist, dass man hier keine freien Texte einfügen muss, da die MARKSTRING-Variante auch 'automatisiert' eingefügte Texte (z.B. COLUMNS LABELS) erkennt und bearbeitet.

Nach demselben Schema wird verfahren, wenn Bilder oder Grafiken auf dem Chart platziert werden sollen⁵. Die POWERCHARTOPTION LabelIconReplace bewirkt, dass bestimmte Textteile durch Bilder ersetzt werden. Die Bilder, die verwendet werden sollen, müssen auf der ersten Seite der PowerPoint-Vorlagedatei abgelegt und können (beliebig) benannt werden⁶.

⁴Siehe [Wildcards in Suchtexten](#) für weitere Optionen zur Definition von Suchtexten.

⁵Siehe hierzu auch POWERCHARTOPTION [LabelIconReplace](#).

⁶Siehe Kapitel 2.

Auf der ersten Folie befinden sich u.a. Icons der fünf Automarken, die im Auswahlbereich benannt wurden.



Abb. 4.8: Bilder auf erster Seite der Vorlagendatei

Möchte man nun die einfache Text-Beschriftung des Balkendiagramms als Text-Tabelle durch die Darstellung des jeweiligen Logos ersetzen, definiert man dies im POWERCHART-Statement mit der folgenden POWERCHARTOPTION:

```
POWERCHARTOPTION "LabelIconReplace=pos=R,resize=Y;
VW Golf=Pic_VW;
Subaru BRZ=Pic_Subaru;
Jaguar F-Type=Pic_Jaguar;
A6=Pic_Audi;
Lada Taiga=Pic_Lada"
```

Hier wird definiert: Ersetze den Text 'VW Golf' durch das Icon namens 'Pic_VW'⁷, den Text 'Subaru BRZ' durch 'Pic_Subaru' usw.! Wenn die Software die entsprechenden Textteile in der Tabelle findet, macht sie genau das: Bild statt Text. Mit `pos=R` wird außerdem eine rechtsbündige Ausrichtung der Icons erzielt. `resize=Y` bewirkt, dass die Icon-Größe an die Zeilenhöhe der Texttabelle angepasst wird.

Das vollständige Statement (ohne die Befüllung des Diagramms, das sollte nun bekannt sein) lautet:

```
POWERCHART
TEMPLATE p
KEY Tab_024
POWERCHARTOPTION "LabelIconReplace=pos=R,resize=Y;
VW Golf=Pic_VW;Subaru BRZ=Pic_Subaru;Jaguar F Type=Pic_Jaguar;
A6=Pic_Audi;Lada Taiga=Pic_Lada"
GRATAB TABLE_01
FIXEDTABDIM
POSITION = | ROWS 1:5 COLUMNS LABELS
;
```

⁷Zum Zwecke der Deutlichkeit wurde hier dieser Bild-Name gewählt — die Benennung der Abbildungen ist aber beliebig und muss kein vorangestelltes 'Pic_' enthalten.

Und das kommt dabei heraus:

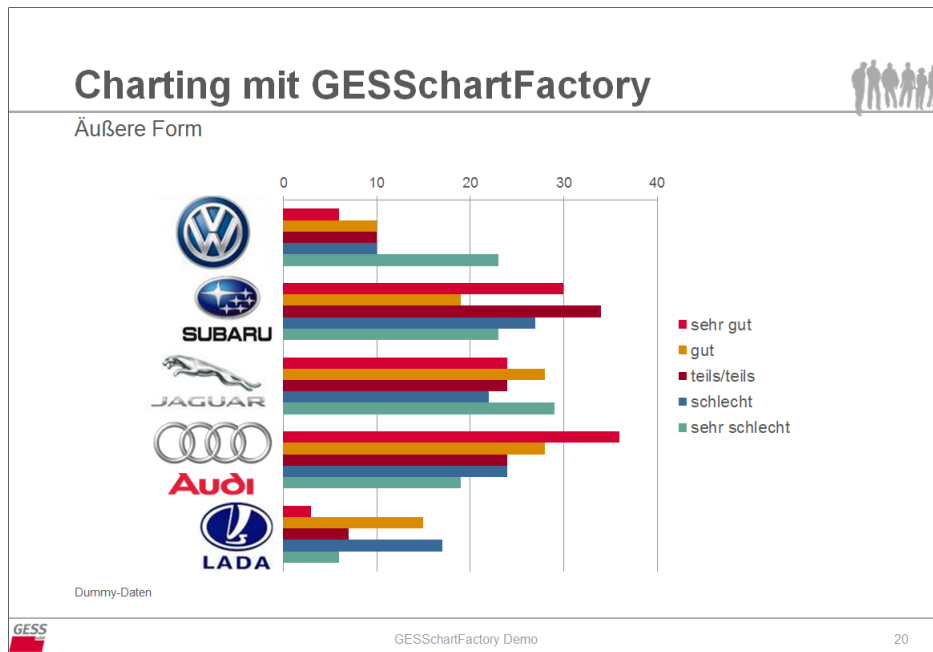


Abb. 4.9: Textersatz mit POWERCHARTOPTION 'LabelIconReplace'

Diese Funktion ist für Tabellen und in leicht abgewandelter Form auch für Textboxen möglich⁸.

⁸Siehe Kapitel 7.3.

5 Liniengrafiken

Gegeben sei eine Tabelle, die Mittelwerte über die Zufriedenheit mit fünf Automarken für sechs Befragten-Gruppen zeigt.

Tab. 5.1: Mittelwerte-Tabelle

GESS ChartFactory Demo					
Abs.	Modellvariante				
	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
Base	116	280	264	272	112
Gruppe 1	3.21	3.01	2.91	3.07	2.68
Gruppe 2	2.72	3.19	3.11	3.19	2.50
Gruppe 3	3.41	2.84	3.18	3.24	2.93
Gruppe 4	3.52	3.20	2.65	3.03	3.00
Gruppe 5	2.93	3.04	2.86	2.78	2.89
Gruppe 6	2.38	3.24	2.98	2.84	2.82

GESS mbH

Für die Darstellung von Mittelwerten bieten sich Liniengrafiken an. Grundsätzlich lassen sich in PowerPoint zwei Arten von Liniengrafiken unterscheiden: Erstens einfache Liniendiagramme und zweitens sogenannte Profil-Diagramme.

5.1 Liniendiagramme

Für erstere stellt PowerPoint eine einfache Diagrammvorlage bereit:

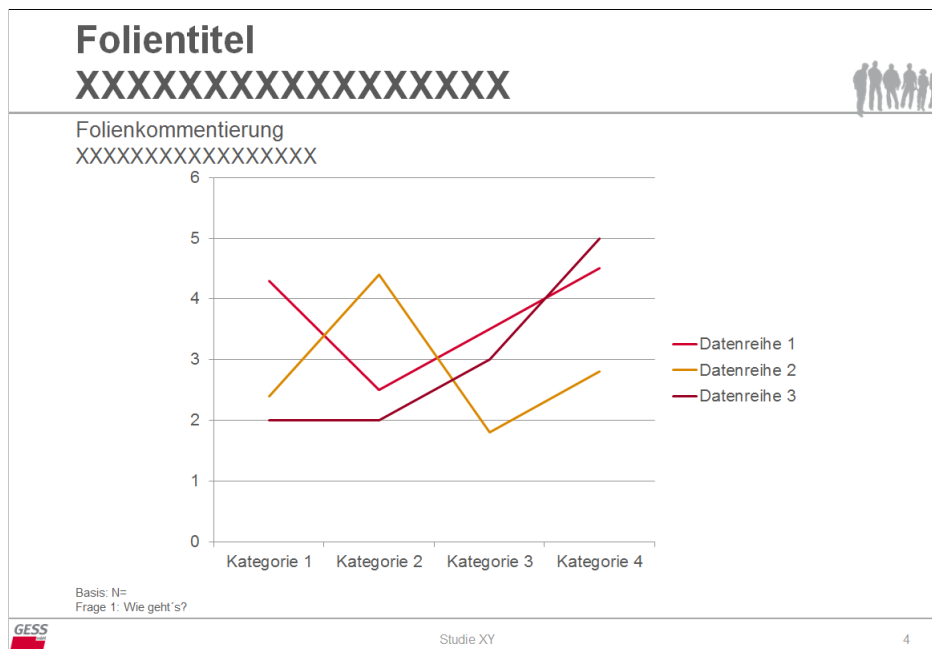


Abb. 5.1: Vorlage für Liniendiagramm

Hier verlaufen die Linien von links nach rechts, die Achsen können optional eingeblendet und beschriftet werden.

	A	B	C	D	E
1		Datenreihe 1	Datenreihe 2	Datenreihe 3	
2	Kategorie 1	4,3	2,4	2	
3	Kategorie 2	2,5	4,4	2	
4	Kategorie 3	3,5	1,8	3	
5	Kategorie 4	4,5	2,8	5	
6					

Das dazugehörige Datenblatt ist übersichtlich und beinhaltet vertikal verlaufende Datenreihen für die horizontal verlaufenden Items.

Abb. 5.2: Datenblatt eines Liniendiagramms

Je nach Darstellungswunsch können die Werte aus der oben stehende Tabelle 1:1 in das Diagramm eingefügt (oder mit dem Statement `INVERSE` transponiert) werden. Hier sollen die Mittelwerte der einzelnen Automodelle als Linien dargestellt werden. Dazu schreibt man einen einfachen `POWERCHART`-Befehl (siehe auch [Beispiel 5: einfaches Liniendiagramm](#)):

```
POWERCHART
TEMPLATE 4
KEY Tab_008
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT ("Zufriedenheit nach Gruppe")
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
POSITION=
| COLUMNS 1:5 ROWS 1:6
;
```

Das Ergebnis sieht folgendermaßen aus:

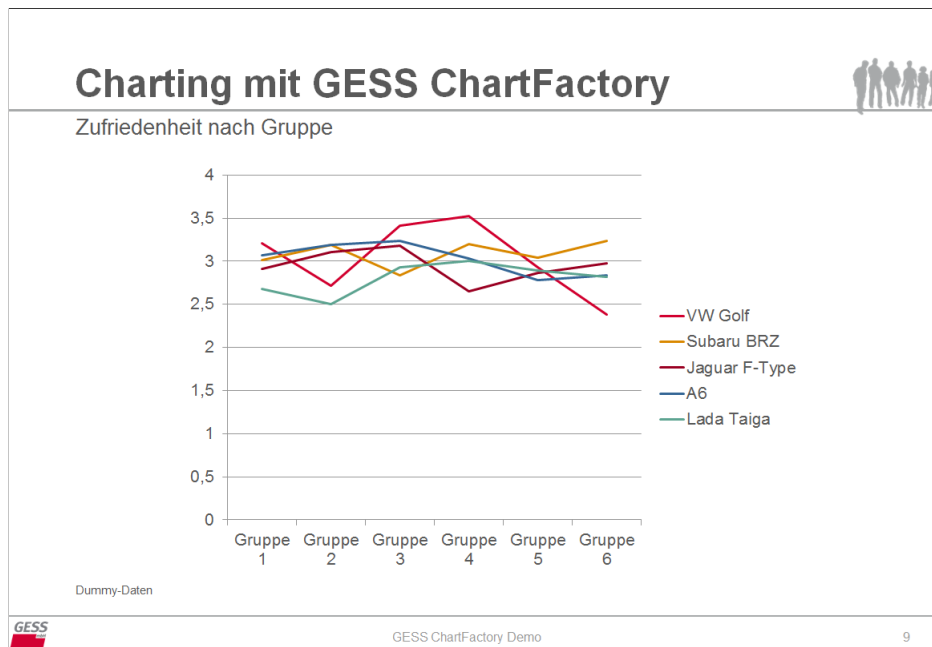


Abb. 5.3: Ausgabe mit Liniendiagramm

5.2 Profildiagramme

Nicht ganz so einfach verhält es sich in dem Falle von Linien, die von oben nach unten verlaufen — sogenannte Profildiagramme.

Hier sieht sich PowerPoint vor Probleme gestellt, die klassischerweise nur auf relativ unkomfortable Weise umschifft werden konnten und zwar mit folgender Vorlage:

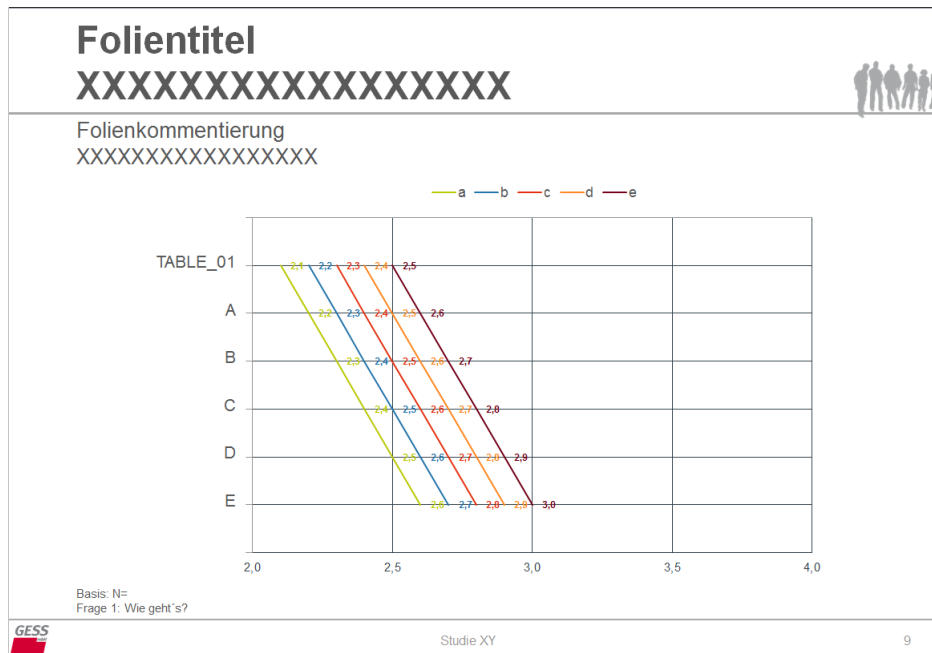


Abb. 5.4: Vorlage für Profildiagramm

... und dem dazugehörigen Datenblatt für das Diagramm:

	A	B	C	D	E	F
1	X-Werte	a	b	c	d	e
2		2,1	1			
3		2,2	2			
4		2,3	3			
5		2,4	4			
6		2,5	5			
7		2,6	6			
8						
9		2,2		1		
10		2,3		2		
11		2,4		3		
12		2,5		4		
13		2,6		5		
14		2,7		6		
15						
16		2,3			1	
17		2,4			2	
18		2,5			3	
19		2,6			4	
20		2,7			5	
21		2,8			6	
22						
23		2,4				1
24		2,5				2
25		2,6				3
26		2,7				4
27		2,8				5
28		2,9				6
29						
30		2,5				1
31		2,6				2

Abb. 5.5: Datenblatt eines Profildiagramms

In der ersten Zeile (ab Zelle B1) stehen die Platzhalter für die Items, hier entsprechend der Vorlage: a bis e. In Spalte A sind die (Dummy-)Werte für die Linien heruntergeschrieben. Die Ziffern in den Spalten B bis F markieren die Positionen der korrespondierenden Werte aus Spalte A auf den jeweiligen Linien der einzelnen Labels. Auf der Linie für Label B werden in der Vorlage beispielsweise sechs Werte aus den Zellen A9 bis A14 dargestellt, deren Position anhand der Ziffern 1-6 in den Zellen C9 bis C14 bestimmt werden. Unbequem (aber machbar) ist die Befüllung dieses Diagramms mit Daten insofern, als dass für jede Datenreihe (jedes Label) neben dem Datenbereich in unserer GESStabs-Tabelle die genaue Startposition im Datenblatt angegeben werden muss.

Das heißt, für fünf darzustellende Linien benötigt man fünf POWERCHART-Anweisungen — oder ein entsprechendes Makro.

Für die erste Linie (Spalte) sieht der Befehl so aus:

```
POWERCHART
KEY Tab_025
TEMPLATE 9
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
```

```
FILL TEXT_02 WITH CONCAT ("Zufriedenheit nach Gruppe")
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
NOLABEL Y NOLABEL X
START A2
FIXEDRANGE
POSITION = | ROWS 1:6 COLUMNS 1
;
```

Labelnamen werden an dieser Stelle nicht eingefügt (`NOLABEL Y NOLABEL X`), die Datenbefüllung startet in Zelle A2 des Datenblattes (`START A2`) und die Dimensionen des Diagrammes sollen beibehalten werden (`FIXEDRANGE`).

Für die restlichen Linien könnte man diesen Befehl viermal kopieren und entsprechend editieren. Alternativ kann mit einem Makro mit drei Parametern gearbeitet werden:

1. `&1` für das Key-Ende
2. `&2` für die Start-Zelle in Spalte A des Datenblattes
3. `&3` für die zu verwendende Spalte in der Datentabelle

```
#macro #Linien (&1 &2 &3)
POWERCHART
KEY TAB_0&1
TEMPLATE p
GRATAB GRAPH_01
NOLABEL X NOLABEL Y
FIXEDRANGE
START A&2
POSITION = | ROWS 1:6 COLUMNS &3
;
#endmacro
```

Dieses wird anschließend in vier kurzen Skriptzeilen aufgerufen, um vier Diagrammlinien zu befüllen:

```
#Linien (26 9 2)
#Linien (27 16 3)
#Linien (28 23 4)
#Linien (29 30 5)
```

Für weitere Ausführungen zur Verwendung von Makros beim Charting siehe Kapitel 6.

Nun fehlen noch zwei Befehle zur Vervollständigung des Charts:

Erstens ein Befehl, mit dem Labelnamen (Automarken) in das Datenblatt integriert und in der Legende dargestellt werden. Da diese nicht zusammenhängend mit den entsprechenden Werten im Datenblatt positioniert sind, kann man sie nicht in einem Rutsch einfügen, sondern muss sie gesondert in die erste Zeile des Datenblattes (Zellen B1 bis F1) einsetzen.

```
POWERCHART
KEY Tab_030
TEMPLATE p
GRATAB GRAPH_01
START B1
```

```
FIXEDRANGE
POSITION = | ROWS LABELS COLUMNS 1:5
;
```

Zweitens folgt eine Anweisung, die die links neben dem Diagramm platzierte Tabelle mit den Labelnamen der Tabellen-Vorspalte (Gruppe 1 - 6) befüllt.

```
POWERCHART
KEY Tab_031
TEMPLATE p
GRATAB TABLE_01
NOLABEL X
FIXEDTABDIM
POSITION = | ROWS 1:6 COLUMNS LABELS
;
```

Das Ergebnis sieht so aus:

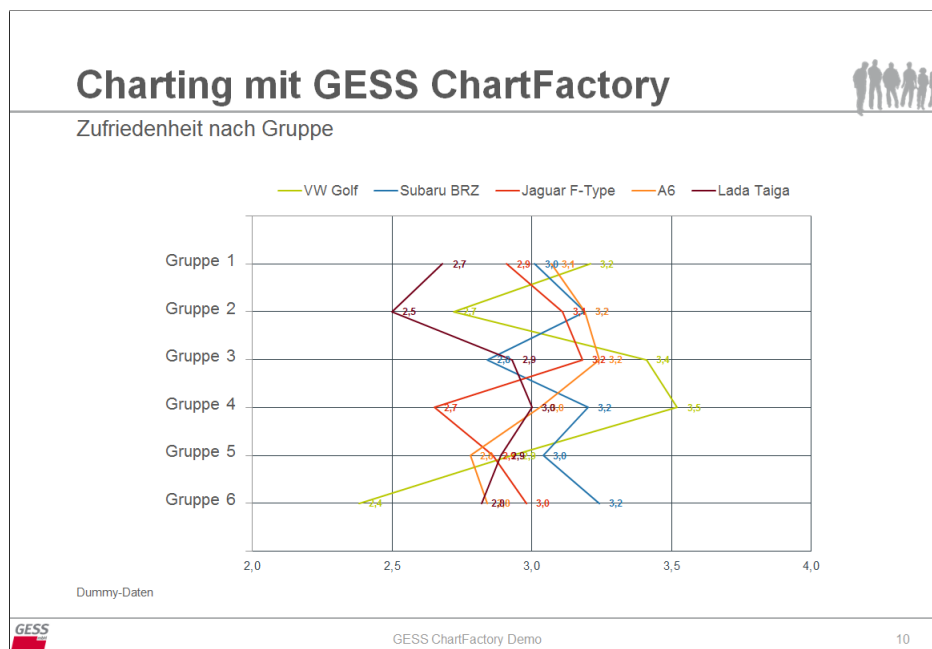


Abb. 5.6: Ausgabe mit Profildiagramm

5.3 GESS-Profilidiagramm

Eine komfortablere (und schnellere) Lösung für ein solches Profildiagramm stellt GESS mit folgender Vorlage bereit.:

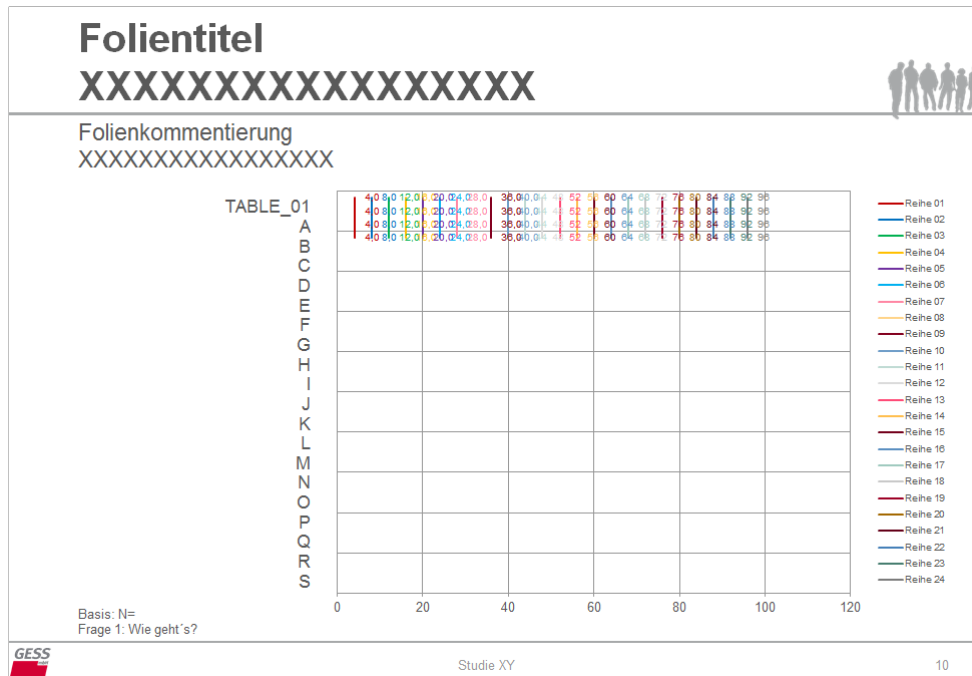


Abb. 5.7: Vorlage für GESS-Profilidiagramm

Der Unterschied besteht vor allem im Aufbau des Datenblattes:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1																										
2	Label 01	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	0,9693
3	Label 02	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	0,95
4	Label 03	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	0,9387
5	Label 04	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	0,9893
6	Label 05	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	0,95
7	Label 06	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	0,9887
8	Label 07																									0,7693
9	Label 08																									0,75
10	Label 09																									0,7887
11	Label 10																									0,6693
12	Label 11																									0,65
13	Label 12																									0,6887
14	Label 13																									0,6693
15	Label 14																									0,55
16	Label 15																									0,5887
17	Label 16																									0,4693
18	Label 17																									0,45
19	Label 18																									0,4887
20	Label 19																									0,3693
21	Label 20																									0,35
22	Label 21																									0,3887
23	Label 22																									0,3693
24	Label 23																									0,25
25	Label 24																									0,2887
26	Label 25																									0,1693
27	Label 26																									0,15
28	Label 27																									0,1887
29	Label 28																									0,0693
30	Label 29																									0,05
31	Label 30																									0,0887
32																										

Abb. 5.8: Datenblatt des GESS-Profilidiagramms

Dieses entspricht dem Aufbau der oben stehenden Datentabelle, die Datenreihen der einzelnen Labels (hier: der sechs Gruppen) verlaufen horizontal. Spalte Z enthält feste Parameter und wird am Ende dieses Kapitels erläutert.

Alle Datenreihen können bequem mit einem Befehl und der POSITION-Angabe = | ROWS 1:6 COLUMNS 1:5 eingefügt werden:

```
POWERCHART
KEY Tab_032
TEMPLATE 10
GRATAB GRAPH_01
NOLABEL X
POSITION = | ROWS 1:6 COLUMNS 1:5
;
```

Die links neben dem Diagramm stehende Tabelle befüllt man, wie zuvor, mit einem separaten POWERCHART-Befehl für die Labelbeschriftung der in der Tabellenseite stehenden Variablen.

```
POWERCHART
KEY Tab_033
TEMPLATE p
GRATAB TABLE_01
NOLABEL X
FIXEDTABDIM
POSITION = | ROWS 1:6 COLUMNS LABELS
;
```

Folgendes Ausgabechart wurde so in nur zwei Schritten und mit einer wesentlich kürzeren Syntax erstellt:

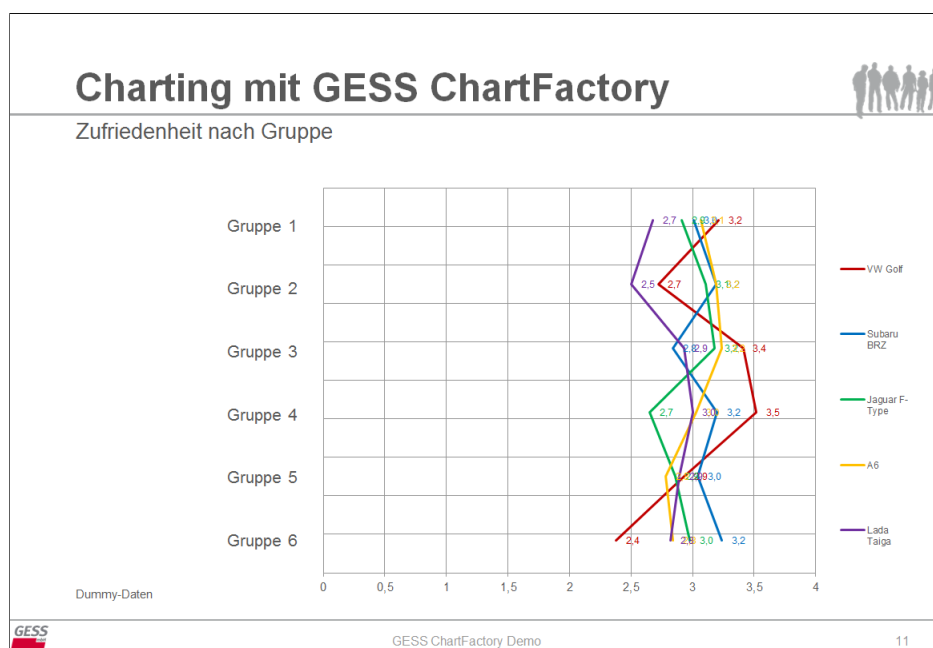


Abb. 5.9: Ausgabe mit GESS-Profilldiagramm

Mit Daten befüllt, sieht das dazugehörige Datenblatt so aus:

	A	B	C	D	E	F	Z
1		VW Golf	Subaru BRZ	Jaguar F-Type A6	Lada Taiga		
2	Gruppe 1	3,21	3,01	2,91	3,07	2,68	0,9167
3	Gruppe 2	2,72	3,19	3,11	3,19	2,5	0,75
4	Gruppe 3	3,41	2,84	3,18	3,24	2,93	0,5833
5	Gruppe 4	3,52	3,2	2,65	3,03	3	0,4167
6	Gruppe 5	2,93	3,04	2,86	2,78	2,89	0,25
7	Gruppe 6	2,38	3,24	2,98	2,84	2,82	0,0833
8	Label 07						0,7833
9	Label 08						0,75
10	Label 09						0,7167

Die Datenstruktur wurde 1:1 aus der Datentabelle übernommen, ebenso wie die Labelnamen.

Abb. 5.10: Befülltes Datenblatt des GESS-Profilldiagramms

Die Parameter in Spalte Z (die nicht besetzten Spalten E bis Y sind hier ausgeblendet) positionieren die Höhe der Datenpunkte auf der Y-Achse und werden beim Übertragen der Excel-Daten automatisch von der Software generiert. Diese Werte dürfen in der Vorlage sowie Ausgabe keinesfalls gelöscht, überschrieben oder verändert werden! Entsprechend kann diese Vorlage für bis zu 30 Items mit je 24 Werten verwendet werden.

Mit der POWERCHARTOPTION AxisPri/-Sec kann die Spannweite von Skalen in Liniengrafiken

beeinflusst werden¹. Auf Abbildung 5.10 sieht man, dass der abgebildete Wertebereich zwischen 2,0 und 4,0 liegt. Möchte man die Punkteverteilung beispielsweise etwas genauer betrachten, kann man mit der `POWERCHARTOPTION AxisPri` einen gewünschten Skalenbereich einstellen. Für X/Y-Graphen werden dafür die sekundären Skalenwerte angesteuert, daher verwendet man in der folgenden Syntaxsequenz für die horizontale Achse das Schlagwort `AxisPR2`.

```
POWERCHART
KEY Tab_032
TEMPLATE 10
GRATAB GRAPH_01
NOLABEL X
POWERCHARTOPTION "AxisPR2=2;4"
POSITION = | ROWS 1:6 COLUMNS 1:5
;
```

Entsprechend wird die Skala auf dem Ausgabechart angepasst:

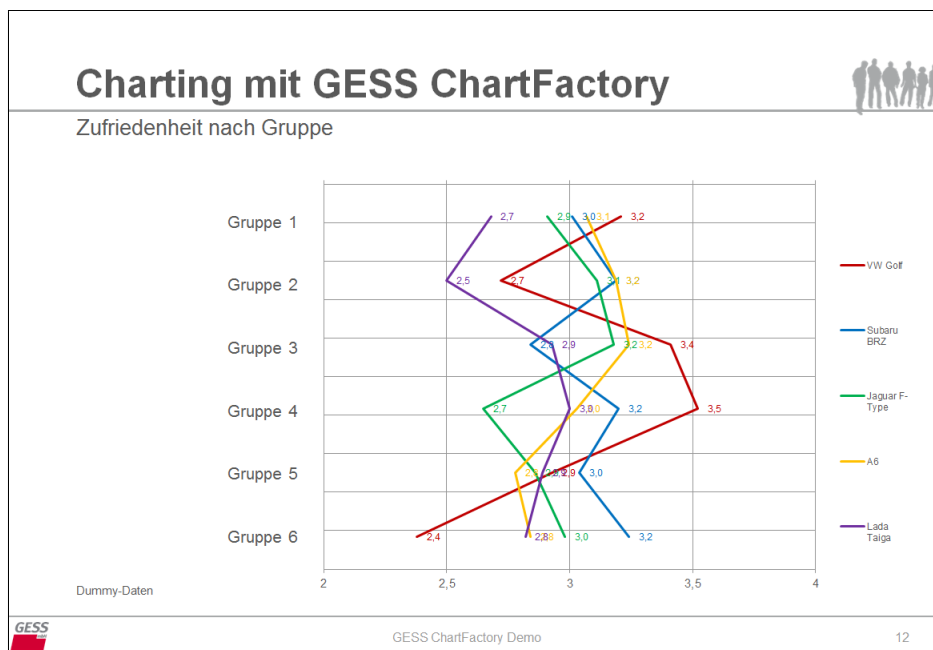


Abb. 5.11: Festlegung eines Achsenbereichs mit `POWERCHARTOPTION 'AxisPri'`

¹Siehe hierzu genauer das Kapitel zu [POWERCHARTOPTION](#).

6 Einbindung von Makros

Die Charterzeugung mit GESS ChartFactory ist syntaktisch eng an die Tabellenerzeugung mit GESSStabs geknüpft. Dies legt nahe, Tabellierung und Charterzeugung in gemeinsame Makros zusammenzufassen — nützlich und häufig verwendet wird dies beispielsweise, wenn viele Messungen auf gleichartigen Skalen vorliegen.

Im Beispiel-Datensatz liegt die Zufriedenheitsbewertung der bekannten Automodellen durch sechs Befragtengruppen vor. Hier die entsprechende Tabelle, beispielhaft für Gruppe 1:

Tab. 6.1: Zufriedenheit mit Automarken, Gruppe 1

GESS ChartFactory Demo						
Col %	Total	Modellvariante				
		VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
Base	1044	116	280	264	272	112
Gruppe 1						
sehr zufrieden	13%	17%	11%	14%	9%	21%
zufrieden	29%	14%	33%	32%	28%	29%
teils/teils	21%	21%	17%	21%	26%	18%
unzufrieden	21%	28%	20%	17%	21%	25%
sehr unzufrieden	16%	21%	19%	17%	16%	7%
sehr zufrieden, zufrieden	42%	31%	44%	45%	37%	50%
unzufrieden, sehr unzufrieden	37%	48%	39%	33%	37%	32%

GESS mbH

Angenommen, man möchte die in der Tabelle unten stehenden Top-2- und Low-2-Box für alle Automodelle und je nach Gruppe auf mit folgendem Vorlagechart auf sechs Ausgabecharts darstellen:

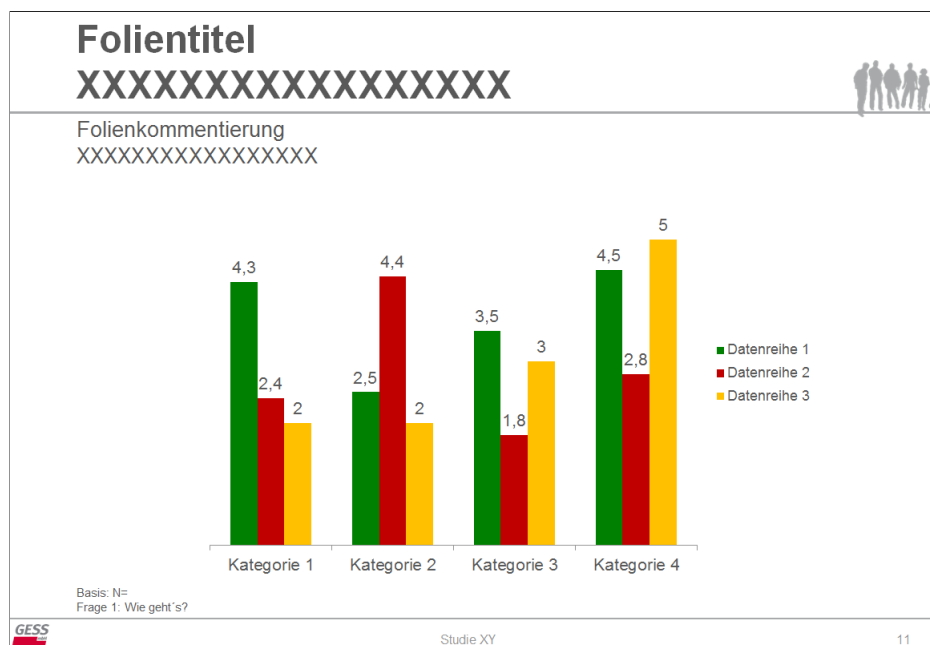


Abb. 6.1: Vorlagechart für Charting-Makro

Anstelle des Schreibens von sechs TABLE- und POWERCHART-Statements bietet es sich an, den für alle sechs Gruppen grundlegend gleichen Befehl in einem Makro zusammenzufassen, das da lautet:

```
#macro #topboxes (&1 &2)
TABLE = k BY &1;
POWERCHART
TEMPLATE 11
KEY &2
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
FILL TEXT_02 WITH CONCAT (VARTITLE &1)
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
INVERSE
POSITION = | ROWS 6 7 COLUMNS 1:5
;
#endmacro
```

Das Makro wird 'Topboxes' genannt. Man benötigt nur zwei Parameter, um das Makro aufzurufen: &1 für den jeweiligen Variablennamen (z1 bis z6), der für die Tabellierung in der ersten Zeile und die Chartbeschriftung in der 7. Zeile benötigt wird, und &2 als Platzhalter für einen, für jedes Statement einzigartigen, KEY.

Alles, was man nun noch tun muss, ist, das Makro sechsmal mit folgenden Inhalten aufzurufen:

```
#topboxes (z1 Tab_034)
#topboxes (z2 Tab_035)
#topboxes (z3 Tab_036)
#topboxes (z4 Tab_037)
#topboxes (z5 Tab_038)
#topboxes (z6 Tab_039)
```

So erstellt man mit sechs kurzen Syntax-Zeilen sechs Charts (und die dazugehörigen Tabellen):



Abb. 6.2: Ausgabe der mit einem Makro erzeugten Charts

Natürlich sind Makros auch weitaus universeller einsetzbar. Mithilfe des folgenden Makros, das nicht grundlos 'autochart' heißt, kann — abgestimmt auf unser Vorlagedesign — nahezu jede beliebige Tabelle und jedes denkbare Chart erzeugt werden:

```
#macro #autochart (&kopf &vorspalte &template &key &element &text &rows &
columns &options)
table = &kopf by &vorspalte;

POWERCHART
TEMPLATE &template
KEY &key
GRATAB &element
FILL TEXT_01 WITH CONCAT ("Charting mit GESS ChartFactory")
```

```

FILL TEXT_02 WITH CONCAT ("%text")
FILL TEXT_03 WITH CONCAT ("Dummy-Daten")
FILL TEXT_04 WITH CONCAT (TABLETITLE)
&options
POSITION = | ROWS &rows COLUMNS &columns
;
#endmacro

```

Die Parameter der Reihe nach:

&kopf	Name des Tabellenkopfes
&vorspalte	Variablenname der Tabellenvorspalte
&template	Chartnummer der Vorlage in der Template
&key	einzigartige Kennung für das Statement
&element	zu füllendes Element auf dem Chart (z.B. GRAPH_01)
&text	beliebiger Text, der den Platzhalter TEXT_02 auf dem Vorlagechart ersetzt
&rows	Ansteuerung der ins Chart zu übernehmenden Tabellenzeilen
&columns	Ansteuerung der ins Chart zu übernehmenden Tabellenspalten
&options	weitere Optionen für das Chart

Dieses Makro kann beispielsweise wie folgt aufgerufen werden:

```

#autochart ( "s" "k" 12 "Tab_040" "GRAPH_01" "Beurteilung der äußeren Form des
VW Golf" 1 "1:5" "INVERSE" )

```

Die Parameter haben dann folgende aktuelle Werte:

&kopf	s (ist eine Variable)
&vorspalte	k (ist eine Variable)
&template	12
&key	Tab_076
&element	GRAPH_01
&text	Beurteilung der äußeren Form des VW Golf
&rows	1
&columns	1:5
&options	INVERSE

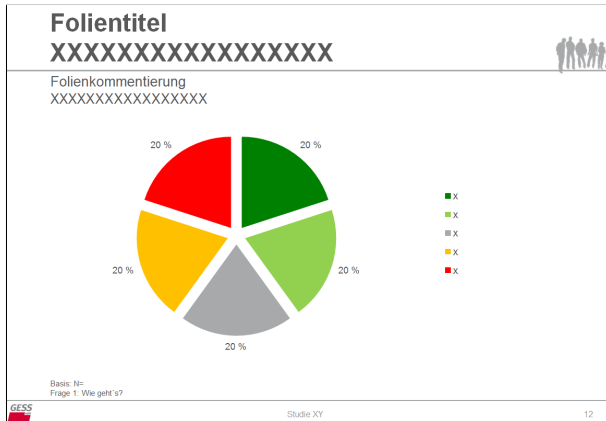


Abb. 6.3: Vorlage für Makro 'autochart'

So werden mit nur einer Skriptzeile und der links stehenden Vorlage die unten stehende Tabelle sowie das folgende Chart erzeugt:

Tab. 6.2: Tabelle für Makro 'autochart'

GESS ChartFactory Demo								
Col %	Total	Äußere Form						
		sehr gut	gut	teils/teils	schlecht	sehr schlecht	Top 2-Box	Low 2-Box
Stufe	1044	132	272	280	235	124	404	380
Modellvariante								
VW Golf	11 %	6 %	10 %	10 %	10 %	23 %	9 %	14 %
Subaru BRZ	27 %	30 %	19 %	34 %	27 %	23 %	23 %	26 %
Jaguar F-Type	25 %	24 %	28 %	24 %	22 %	29 %	27 %	24 %
Audi	26 %	36 %	28 %	24 %	24 %	19 %	31 %	22 %
Lada Tags	11 %	3 %	15 %	7 %	17 %	6 %	11 %	13 %

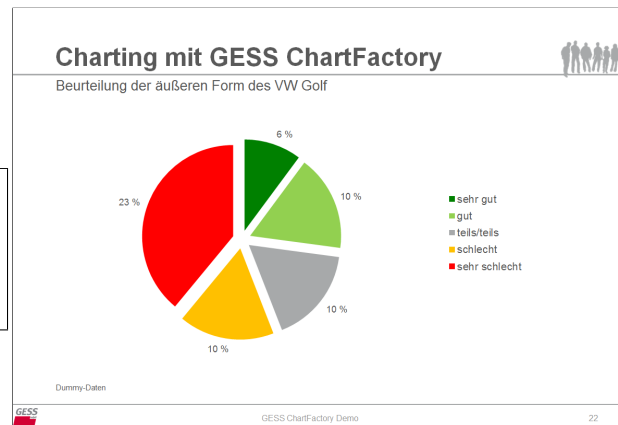


Abb. 6.4: Ausgabe mit Makro 'autochart'

Auf Grundlage eines Makros können mit einer Skript-Zeile beliebig viele Charts und Charttypen generiert werden. Mehr Aufwand muss nicht sein.

Gewiss ist der Einsatz von Makros nicht an jeder Stelle nötig oder sinnvoll. Ein solches Makro macht es bspw. möglich und einfach, die Einzelheiten der Tabellen- und Charterzeugung vor dem Benutzer zu verbergen.

7 POWERCHARTOPTION

Mittels POWERCHARTOPTION können Diagramme und Tabellen anhand ihrer Inhalte (Texte oder Zahlenwerte) optisch beeinflusst und gesteuert werden. Bevor die einzelnen Optionen erläutert werden, ein kurzer Überblick über deren Anwendungsmöglichkeiten:

Tab. 7.1: Anwendungsgebiete der POWERCHARTOPTIONs

Option	TABLE	GRAPH
Markstring	✓	✗
Transpon	✓	✓
LabelIconReplace	✓	✗
XYScatterIcon	✗	✓
BaseCheck	✓	✓
Sort	✓	✓
ValueDiffTo[Max Min]	✓	✗
Formula	✗	✓
DataFromGraph	✓	✗
AxisPri	✗	✓
CellColor	✓	✗
CellFontColor	✓	✗
CellIcon	✓	✗
Per[Row Col]	✓	✗
SeriesColor	✗	✓
SeriesLabelColor	✗	✓
SeriesColorMarkstring	✗	✓
SeriesLabelAdd	✗	✓
SeriesLabelReplace	✗	✓
FillColorObject	✓	✓
DeleteObject	✓	✓

Die allgemeine Syntax lautet POWERCHARTOPTION "Optionen" und wird einfach in den POWERCHART-Befehl integriert. Ein Befehl kann beliebig viele Optionen enthalten, die mit Pipe (|) getrennt werden.

Eine typische Syntaxfrequenz lautet zum Beispiel:

```
POWERCHART
TEMPLATE 2
KEY Tab_001
GRATAB GRAPH_01
FILL TEXT_01 WITH CONCAT (TABLETITLE)
POWERCHARTOPTION "BaseCheck=(Base <= 200);Remove;Italic;Bottomtext=TEXT_06=
Achtung, Basis zu gering|LabelIconReplace=pos=R,resize=Y,A6=Pic_AUDI|SORT=C|
Cellcolor=value<=10;$00FF00|PerCol"
POSITION = | ROWS 1:5 COLUMNS 1:5
;
```

7.1 Markstring

Syntaxstruktur:

```
Markstring=<Text{,LINE}{,ARRANGE=F|B}>{;MARKBOLD}; [FOREGROUND=<Farbcode> /
BACKGROUND=<Farbcode>]
```

Mit `Markstring` können Texte angesprochen werden, die innerhalb von PowerPoint-Tabellen in Fettschrift markiert und/ oder eingefärbt werden sollen.

Folgende Optionen stehen bei greifender Markierung zur Verfügung:

LINE	Markierung der gesamten Tabellenzeile
ARRANGE F B	Verrücken der Tabelle in den Vorder- oder Hintergrund (F=Front, B=Back)
MARKBOLD	Fettschrift
FOREGROUND <Farbcode>	Schriftfarbe, zu kennzeichnen mit Hexadezimal- oder RGB-Code
BACKGROUND <Farbcode>	Hintergrundfarbe, ebenfalls via Hexadezimal- oder RGB-Code

Wenn der exakte Text, der markiert werden soll, nicht bekannt ist oder mehrere Texte mit identischen Textteilen erkannt werden sollen, können Textteile mit Wildcards gesucht werden:

Tab. 7.2: Wildcards in Suchtexten

Ausdruck	Beispiel: [Seriescolor]Markstring=	Erklärung
*	*Golf Golf* *Golf*	Das Sternchen dient als Platzhalter für Wörter vor und/oder nach dem definierten Text.
?	VW G??f VW Go?f	Das Fragezeichen dient als Platzhalter für (unbekannte) Buchstaben innerhalb von Wörtern. Der Suchtext wird erkannt, wenn die angegebene Anzahl an Fragezeichen der tatsächlichen Anzahl an Buchstaben entspricht.
[x]	G[ol][ol]f V[WHBTM]	Der Suchtext wird erkannt, wenn die in Klammern angegebenen Buchstaben an der entsprechenden Stelle des Textes stehen. Im ersten Beispiel darf an 2. und 3. Stelle jeweils ein O oder L stehen.

7.2 Transpon

Syntaxstruktur:

`Transpon`

Analog zu `INVERSE` transponiert `Transpon` die Achsen eines angesprochenen Datenbereichs in Tabellen und Diagrammen. Anders als `INVERSE`¹ erfolgt mit `Transpon` auch eine physische Drehung der übertragenen Datenbereiche innerhalb von Diagramm-Datenblättern.

¹`INVERSE` bewirkt für Diagramme lediglich eine umgedrehte Darstellung des Datenbereichs, überträgt die Datenreihen jedoch in ursprünglicher Form in das Diagramm-Datenblatt.

Im unten stehenden Beispiel wird so bewirkt, dass die Achsen der GESStabs-Tabelle physisch gedreht und in das Datenblatt des PowerPoint-Diagramms eingefügt werden.

POWERCHARTOPTION "Transpon"

GESStabs-Tabelle						
GESStabs-Tabelle						
Col %	Base	Modellvariante				
		VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
Äußere Form						
sehr gut	132	7 %	14 %	12 %	18 %	4 %
gut	272	24 %	19 %	29 %	28 %	36 %
teils/teils	280	24 %	34 %	26 %	25 %	18 %
schlecht	236	21 %	23 %	20 %	21 %	36 %
sehr schlecht	124	24 %	10 %	14 %	9 %	7 %
Top 2-Box	404	31 %	33 %	41 %	46 %	39 %
Low 2-Box	360	45 %	33 %	33 %	29 %	43 %

Abb. 7.1: POWERCHARTOPTION 'Transpon (GESStabs-Tabelle)'

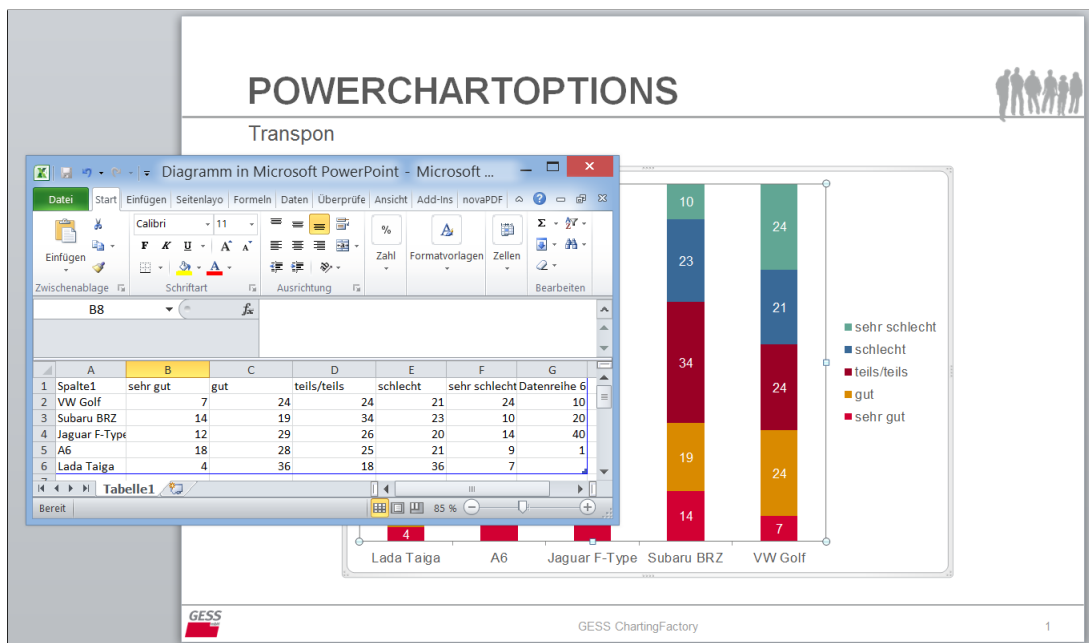


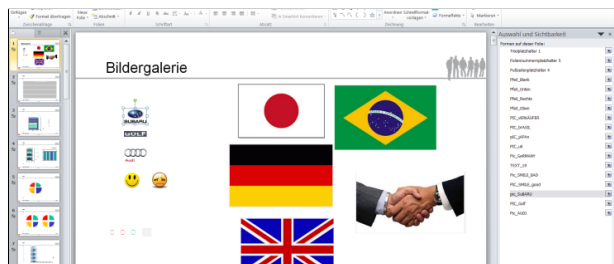
Abb. 7.2: POWERCHARTOPTION 'Transpon'

7.3 LabelIconReplace

Syntaxstruktur:

LabelIconReplace=pos=L|C|R{,resize=X|Y|B};<Text>=<Bild>

Mit LabelIconReplace können Texte innerhalb von Tabellen und Textboxen durch Bilder ersetzt werden.



Hierfür müssen die Bilder auf der **ersten** Seite der PPT-Vorlagendatei mit den gewünschten Element-Namen definiert sein. Die zu platzierenden Bilder können in ihrer Größe und Position innerhalb der Tabellenzelle beeinflusst werden.

Abb. 7.3: Bilder auf erster Seite der Vorlagendatei

Mit `pos` kann die Position des Bildes innerhalb des PowerPoint-Elementes² gesteuert werden. Mit `resize` kann die Größe (abhängig von der Bildeinstellung in PowerPoint) der Grafik angepasst werden — Achtung: die Option 'Seitenverhältnis sperren', die PowerPoint in den Bildformateinstellungen bereitstellt, darf zu diesem Zwecke nicht aktiviert sein! Ein entsprechender Hinweis ist unter [Deaktivieren von 'Seitenverhältnis sperren'](#) zu finden.

Die Einstellungen zur Größenanpassung ist optional, die jeweiligen Parameter bewirken folgende Formatierungen:

Tab. 7.3: Optionen für POWERCHARTOPTION 'LabelIconReplace'

<code>pos={L C R}</code>	L = Left C = Center R = Right
<code>resize={X Y B}</code>	X = Zellenbreite Y = Zellenhöhe B = Both (Breite + Höhe)

Alle Texte, welche innerhalb von Tabellen durch Bilder ersetzt werden sollen, müssen ihres Wortlautes nach oder mithilfe von Wildcards³ im Statement aufgeführt werden.

Ein typischer Befehl lautet zum Beispiel:

```
POWERCHARTOPTION "LabelIconReplace=pos=R,resize=Y;VW Golf=Pic_VW;Subaru BRZ=Pic_Subaru;Jaguar F-Type=Pic_Jaguar;A6=Pic_Audi;Lada Taiga=Pic_Lada"
```

Für den Einsatz von `LabelIconReplace` in einer Textbox wird das Statement `FILL TEXT_[xx] WITH FREETEXT` verwendet. Hier braucht kein zu ersetzender Text, sondern nur das einzusetzende Bild genannt werden. Es stehen die gleichen optionalen Formateinstellungen zur Verfügung.

Beispiel-Befehl: `FILL TEXT_05 WITH FREETEXT "LabelIconReplace=pos=c,resize=B;Pic_Germany"`.

²D.h. innerhalb einer Tabellenzelle oder Textbox

³Siehe 7.2

Aus einer Kombination dieser beiden POWERCHARTOPTION-Anweisungen und der links stehenden Vorlage ergibt sich dann folgendes Chart:

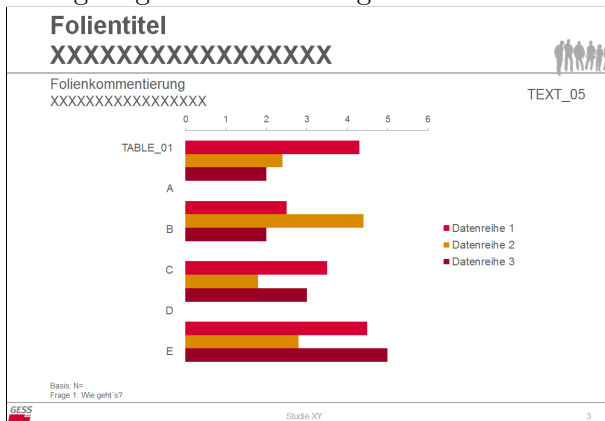


Abb. 7.4: Vorlage für LabelIconReplace

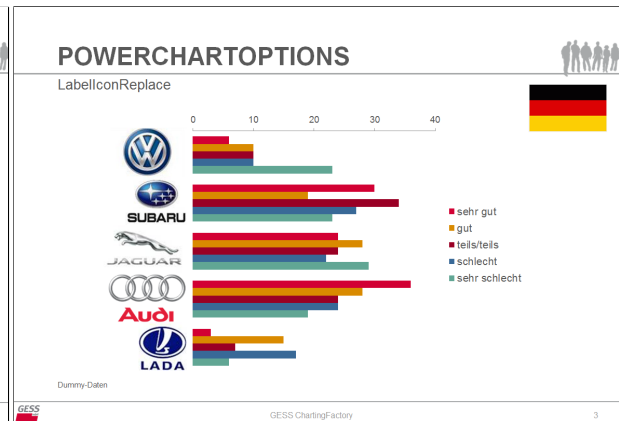


Abb. 7.5: LabelIconReplace

7.4 XYScatterIcon

Syntaxstruktur:

`XYScatterIcon=pos=L|R|T|B{, arrange=F|B}; <Text>=<Bild>`

Analog zu `LabelIconReplace` können mit `XYScatterIcon` Texte durch Bilder ersetzt werden, hier in XY-Diagrammen.

Die Bilder werden in ihrer Originalgröße in das Linien-, Profil- oder Punktediagramm eingefügt. Mit `pos` kann das Bild um den Datenpunkt herum positioniert werden, `arrange` platziert es im Vorder- oder Hintergrund. Wird keine Angabe für `arrange` gemacht, erscheint das Bild im Vordergrund. Nachstehend sind die möglichen Parameter für beide Optionen aufgeführt:

Tab. 7.4: Optionen für POWERCHARTOPTION 'XYScatterIcon'

<code>pos=L R T B</code>	<p>L = links (left)</p> <p>R = rechts (right)</p> <p>T = über (top)</p> <p>B = unter (bottom)</p>
<code>arrange=F B</code>	<p>F = Vordergrund (Foreground)</p> <p>B = Hintergrund (Background)</p>

```
POWERCHARTOPTION "XYScatterIcon=pos=R;VW Golf=Pic_VW_mini|XYScatterIcon=pos=L;
Lada Taiga=Pic_Lada_mini"
```

Mit dem oben stehenden Befehl wird z.B. folgende Ausgabe erzielt:

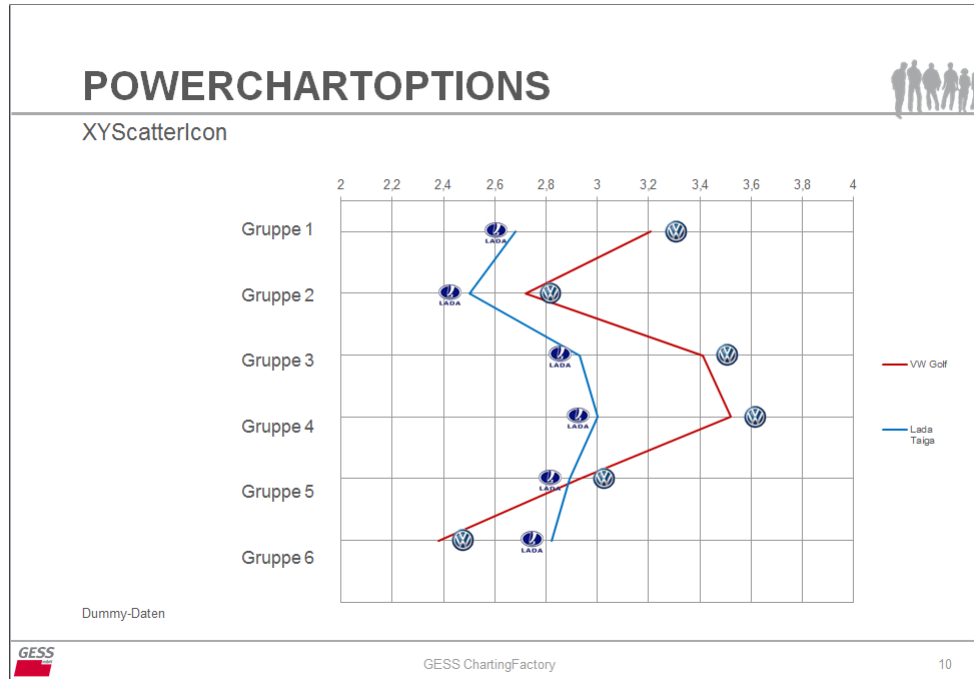


Abb. 7.6: POWERCHARTOPTION 'XYScatterIconreplace'

7.5 BaseCheck

Syntaxstruktur:

```
BaseCheck = (Bedingung); {Remove|Italic|NoDataVal|NoDataChar|NoLineFeed};
Bottomtext=<TEXT_[xx]>=<Hinweistext>
```

Mit **BaseCheck** können Basen, die den in einem Diagramm oder einer Tabelle dargestellten Datenreihen zugrunde liegen, überprüft und bei Abweichungen vom definierten Wertebereich markiert werden.

Der gewünschte Wertebereich wird anhand von Schwellenwerten definiert:

Tab. 7.5: Definition des Wertebereichs für POWERCHARTOPTION 'BaseCheck'

BaseCheck = (Base <= x)	Wertebereich unter/ über Zahlenwert
BaseCheck = (Base >= x)	Wertebereich unter/ über Zahlenwert
BaseCheck = (Base >= x and Base <=y)	Wertebereich zwischen zwei Zahlenwerten
BaseCheck = (Base <= x or Base >=y)	Nicht aneinander grenzende Wertebereiche

Die Möglichkeiten zur Kennzeichnung abweichender Basen sind optional.

Markiert werden können zu hohe/ niedrige Basen einerseits durch einen Hinweistext, der als Fußnote (automatisch: hochgestellte 3⁴) an die entsprechenden Labels und Zählwerte geschrieben wird:

```
Bottomtext=<TEXT_[xx]>=<Hinweistext>.
```

⁴In der Powerchart.xlsb kann das Fußnoten-Symbol in Zelle A7 nach 'BASENOTECHAR=' beliebig geändert werden.

Außerdem können die entsprechenden Labels und/ oder Zählwerte innerhalb von Diagrammen bei vom gewünschten Bereich abweichenden Basen formatiert werden. Folgende Optionen sind möglich und (mit Semikolon getrennt) beliebig kombinierbar:

Tab. 7.6: Optionen für POWERCHARTOPTION 'BaseCheck'

Remove	Entfernt die Basen-Angabe im Labeltext
Italic	Label und/oder Zählwert in Kursiv
NoDataVal	Zählwerte werden nicht gezeigt
NoDataChar	Fußnotensymbol (³) wird nicht gezeigt

Die Formatanweisungen greifen nur mit folgender Formatdefinition der in den Labeltexten eingblendeten Basen: `PPNUMFORMAT = '|| (n=#)'`; . Der durch `||` erzeugte Zeilenumbruch zwischen Labeltext und `(n=x)` kann mit der Option `NoLineFeed` innerhalb des `BaseCheck`-Befehls wieder entfernt werden.

Das Überprüfen und Markieren zu niedriger Basen kann dann beispielsweise so erfolgen:

```
POWERCHARTOPTION "BaseCheck=(Base <= 200);Remove;Italic;Bottomtext=TEXT_06=
Achtung, Basis zu gering"
```

Hier werden Basen unter `N=200` mit einer Fußnote markiert, die Basenanzeige in den Labeltexten wird deaktiviert, alle betreffenden Label- und Datenreihentexte werden kursiv dargestellt und in der Textbox namens `TEXT_06` wird der Hinweistext '³ Achtung, Basis zu gering' angezeigt.

Auf den nachstehenden Abbildungen sind verschiedene Varianten der `BaseCheck`-Option in Diagrammen- und Tabellenelementen dargestellt.

Diagramm:

POWERCHARTOPTION "BaseCheck=(Base<=200);NoLineFeed;Bottomtext=TEXT_05=Achtung, Basis zu gering"

Tabelle:

POWERCHARTOPTION "BaseCheck=(Base<=200);Italic;Bottomtext=TEXT_05=Achtung, Basis zu gering"

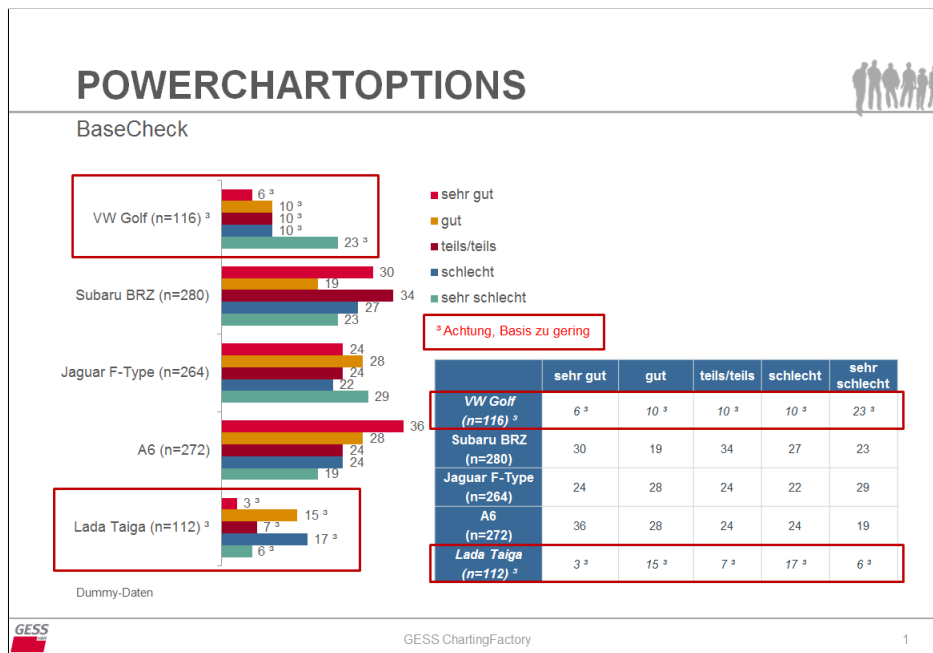


Abb. 7.7: Varianten von POWERCHARTOPTION 'BaseCheck'

Diagramm:

```
POWERCHARTOPTION "BaseCheck=(Base<=200);NoLineFeed;NoDataVal;Bottomtext=TEXT_05=Achtung, Basis zu gering"
```

Tabelle:

```
POWERCHARTOPTION "BaseCheck=(Base<=200);NoDataVal;NoDataChar;Bottomtext=TEXT_05=Achtung, Basis zu gering"
```

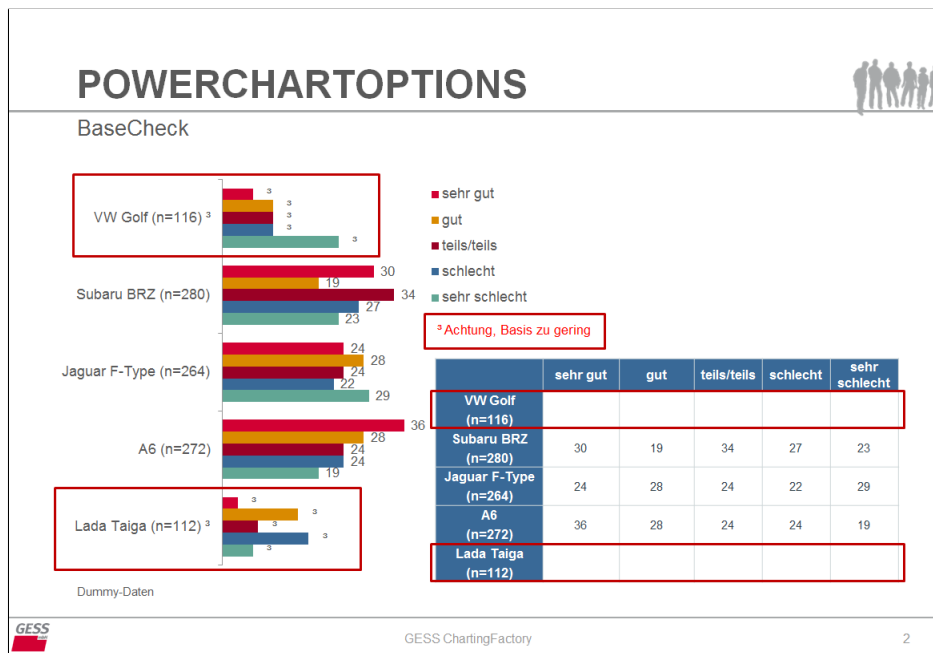


Abb. 7.8: Weitere Varianten von POWERCHARTOPTION 'BaseCheck'

7.6 Sort

Syntaxstruktur:

Sort=<Tabellenspalte>

Mit Sort können Datenreihen innerhalb einer Excel-Datentabelle nach einer Spalte (Absolut-Spalte im Diagramm- oder Tabellen-Datenblatt) sortiert werden, wenn bestimmte Sortierungen in GESStabs selbst nicht möglich oder gewünscht sind. Standard ist eine absteigende Sortierung (DESCENDING), durch Ergänzen von ;ASCENDING erfolgt eine aufsteigende Sortierung.

In der unten stehenden Tabelle wurden die Skalenpunkte absteigend nach Spalte C (im Datenblatt) sortiert: `POWERCHARTOPTION "SORT=C"` .

POWERCHARTOPTIONS

SORT

	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
gut	24	19	29	28	36
teils/teils	24	34	26	25	18
schlecht	21	23	20	21	36
sehr schlecht	24	10	14	9	7
sehr gut	7	14	12	18	4

Dummy-Daten

GESSESS ChartingFactory 4

Abb. 7.9: POWERCHARTOPTION 'Sort'

7.7 ValueDiffToMax/ ValueDiffToMin

Syntaxstruktur:

`ValueDiffToMax|ValueDiffToMin`

Mit `ValueDiffToMax` bzw. `ValueDiffToMin` können die Zählwerte in Tabellenzellen als Abweichung vom größten oder kleinsten Wert der Datenreihe berechnet werden. Mit der `POWERCHARTOPTION "ValueDiffToMax"` wird der Maximalwert der Tabelle vom jeweiligen Zellen-Wert subtrahiert: Neuer Zell-Wert = Zellen-Wert – Max-Wert. `ValueDiffToMin` wird analog berechnet, hier wird der kleinste Wert subtrahiert.

Zum Beispiel:

Zell-Wert = 70

Min = 20

Max = 90

ValueDiffToMax: Zell-Wert – Max = Neuer Zell-Wert 70-90 = -20

ValueDiffToMin: Zell-Wert – Min = Neuer Zell-Wert 70-20 = 50

Das unten stehende Chart bildet zwei Tabellen ab, in deren Zellen der kleinste (oben) und der größte (unten) Wert der Datentabelle abgezogen wurde.

POWERCHARTOPTIONS						
ValueDiffToMin/Max						
	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga	
sehr gut	3	10	8	14	4	
gut	20	15	25	24	32	
teils/teils	20	30	22	21	14	
schlecht	17	19	16	17	32	
sehr schlecht						
	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga	
sehr gut	-29	-22	-24	-18	-32	
gut	-12	-17	-7	-8	36	
teils/teils	-12	-2	-10	-11	-18	
schlecht	-15	-13	-16	-15	36	
sehr schlecht	-12	-26	-22	-27	-29	

Dummy-Daten

GESS ChartingFactory 5

Abb. 7.10: POWERCHARTOPTION 'ValueDiffToMax/ Min'

So können aus den Zahlenwerten einer Datentabelle PowerPoint-Tabellen mit verschiedenen Werten berechnet werden.

7.8 Formula

Syntaxstruktur:

Formula==Formel;Start=<Startzelle>;{Range=<Tabellenbereich>}

Mit Formula können Excel-übliche Formeln in das Datenblatt von Diagrammen eingefügt werden. Mit Start=<Startzelle> wird die Startzelle definiert. Optional kann mit Range=<Tabellenbereich> außerdem ein Tabellenbereich angegeben werden, in den die Formel eingefügt bzw. fortgesetzt werden soll.

Zum Beispiel fügt die folgende POWERCHARTOPTION-Anweisung erstens eine Formel zur Beschriftung der neu generierten Zeile (in diesem Fall enthält die 'Formel' nur den gewünschten Labeltext) und zweitens eine Formel zur Berechnung einer Low-2-Box (Kategorien 'schlecht' + 'sehr schlecht') in Zeile 7 ein:

```
POWERCHARTOPTION "Formula=='Zeile aus Formel berechnet';START=A7|Formula==B5+B6;Start=B7;Range=B7:F7"
```

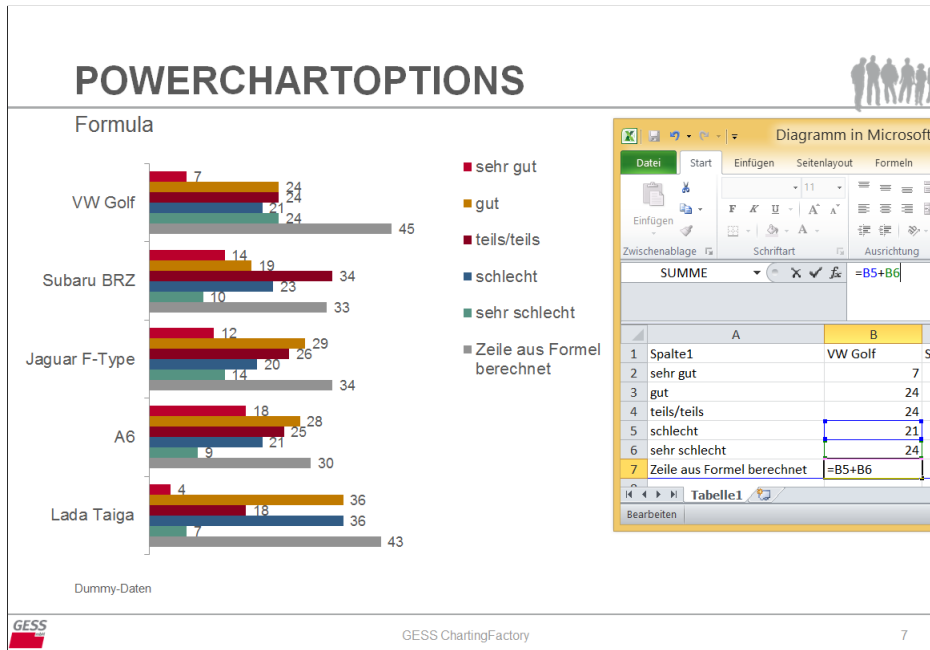


Abb. 7.11: POWERCHARTOPTION 'Formula'

Die Tabellenbereiche, die mit anhand von Formeln errechneten Werten gefüllt werden sollen, müssen im POSITION-Statement des POWERCHART-Befehls berücksichtigt werden.

Wenn im Voraus nicht bekannt ist, wie viele Zeilen/ Spalten ausgegeben werden, aber eine Referenz für Formelangaben und/ oder den Zellenbereich angegeben werden soll, können hilfsweise folgende Syntaxsequenzen verwendet werden:

Tab. 7.7: Referenzangaben bei unbekannter Tabellengröße für POWERCHARTOPTION 'Formula'

#RL- 9#	...	#RL0#	...	#RL+9#	,letztgültige' Zeile (Row Last)
#CL- 9#	...	#CL0#	...	#CL+9#	,letztgültige' Spalte (Column Last)

#RL-5# bedeutet zum Beispiel, dass fünf Zeilen vor der letztgültigen Zeile, #CL+3# beispielsweise, dass drei Spalten nach der letztgültigen Spalte verwendet werden sollen.

In Formeln notwendige Anführungsstriche (") müssen als einfache Anführungszeichen (') gesetzt werden.

Beachte: Formula ist ein äußerst mächtiges und effektvolles Feature zur Datenberechnung und -manipulation. Die errechneten Werte können im Nachhinein nur über das Skript und im Excel-Datenblatt nachvollzogen werden, da sie nicht nachträglich in die GESSstabs-Tabelle übertragen werden. Wenn möglich, wird eine kontrolliertere Form der Datenberechnung in GESSstabs (beispielsweise mittels COMPUTE oder CALCULATECOLUMN) empfohlen!

7.9 DataFromGraph

Syntaxstruktur:

`DataFromGraph=GRAPH_[xx];<Zellenbereich>`

Mit `DataFromGraph` können Daten aus dem Datenblatt eines auf dem Chart dargestellten Diagramms übernommen und in einer Tabelle auf demselben Chart abgebildet werden. Dies kann gegebenenfalls nötig werden, wenn in dem Diagramm durch Formeln oder ähnliches weitergehende Wertberechnungen durchgeführt und diese anschließend (zusätzlich) in einer Tabelle gezeigt werden sollen.

In diesem Beispiel wird mit `POWERCHARTOPTION "DataFromGraph=GRAPH_01;A1:F6"` der rot markierte Datenbereich (Zelle A1 bis F6) aus dem Datenblatt des dargestellten Diagramms ausgewählt und in die nebenstehende Tabelle eingefügt⁵.

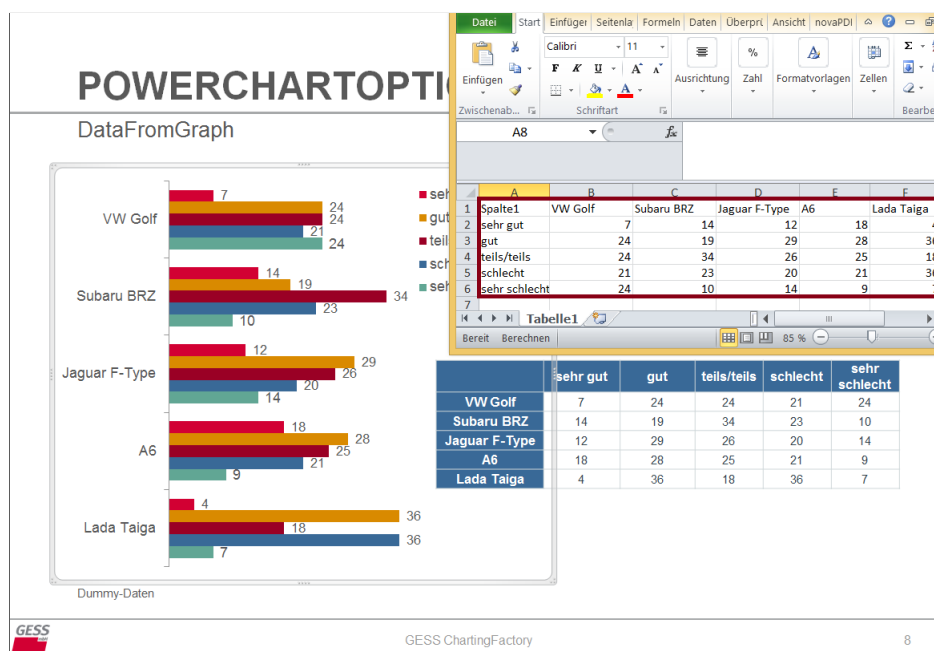


Abb. 7.12: POWERCHARTOPTION 'DataFromGraph'

Ist im Vorneherein nicht bekannt, wie groß das zu übertragende Datenblatt ist, und demnach kein genauer Tabellenbereich angegeben werden kann, kann auf die [Referenzangaben bei unbekannter Tabellengröße für POWERCHARTOPTION 'Formula'](#) zurückgegriffen werden.

7.10 AxisPri

Syntaxstruktur:

`AxisPr[i|2]= [<Skala-Min>;] <Skala-Max>`

`AxisSe[c|2]= [<Skala-Min>;] <Skala-Max>`

Standardmäßig⁶ wird die Achsenskalierung von Diagrammen bei der Datenübertragung in PowerPoint automatisch an den Wertebereich der darzustellenden Daten angepasst. Mit `AxisPr`

⁵Damit die Automarken (wie im Diagramm) links stehen, wurden Zeilen und Spalten in der Tabellenansicht (gegenüber des Diagramm-Datenblattes) transponiert.

⁶Wenn Sie keine feste Skalierung in PowerPoint gewählt haben.

[i|2] und AxisSe[c|2] können die Skalenbereiche der primären und sekundären Achse eines Diagramms manuell angepasst werden.

Die vertikale Primärachse eines Diagramms wird mit `POWERCHARTOPTION "AxisPri=20;100"` beispielsweise auf den Wertebereich 20 bis 100 skaliert. `AxisSec` spricht die horizontale Primärachse an.

Auf den Befehl hin prüft das Programm, ob alle darzustellenden Daten innerhalb des angegebenen Achsenbereichs liegen. Wenn nicht, wird ein Warnhinweis auf dem Chart ausgegeben.

In dem unten stehenden Beispiel liegt etwa der kleinste mit dem `POSITION`-Statement angesprochene Wert nicht innerhalb des definierten Achsen-Bereiches (`AxisSec=2,5;4`).

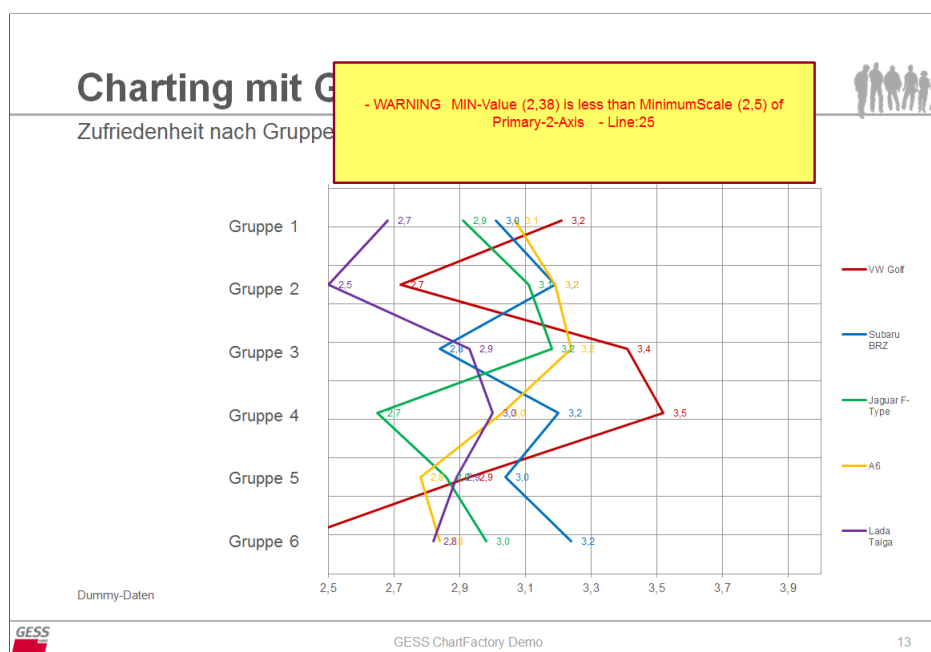


Abb. 7.13: POWERCHARTOPTION 'AxisPri' mit Warnhinweis

Wird nur ein Wert angegeben, wird dieser als Maximalwert interpretiert und das Skalen-Minimum der Chartvorlage nicht verändert.

Analog können mit `AxisPr2` und `AxisSe2` die sekundären Skalenwerte der vertikalen und horizontalen Achse bei X/Y-Graphen gesteuert werden.

7.11 CellColor

Syntaxstruktur:

`CellColor=<Bedingung>;<Farbangabe>`

Mit `CellColor` können Tabellenzellen in Abhängigkeit von ihren Zahlenwerten farbig hinterlegt werden. Die jeweiligen Zahlenwerte können auf unterschiedliche Art anhand von Bedingungen geprüft werden. Der gewünschte Farbcode kann entweder über die bekannte Hexadezimal-Notation `$1122AA`/`#1122AA` oder mit einem RGB-Code `xxx,xxx,xxx` angegeben werden.

Beispielsweise werden mit folgendem Befehl alle Werte unter 11 in der Tabelle in Signalgelb markiert: `POWERCHARTOPTION "CellColor=value<=10;$00FF00"`

POWERCHARTOPTIONS

Cellcolor

	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
sehr gut	7	14	12	18	4
gut	24	19	29	28	36
teils/teils	24	34	26	25	18
schlecht	21	23	20	21	36
sehr schlecht	24	10	14	9	7

Dummy-Daten

GESS ChartingFactory 10

Abb. 7.14: POWERCHARTOPTION 'CellColor'

Die farbliche Markierung kann anhand folgender Bedingungen gesteuert werden:

Tab. 7.8: Formulierung von Bedingungen für POWERCHARTOPTION 'CellColor'

Ausdruck	Beispiel: Cellcolor=	Erklärung
> , < , >= , <=	value>=10;\$ADADEB	Zellen mit für die Bedingung zutreffenden Zahlenwerten werden mit der gewünschten Farbe hinterlegt.
<Zelle> + - * /	value >= C1+5;\$ADADEB	Hier wird der aktuelle Zellenwert gegen einen anderen Zellenwert aus (absoluter, inkl. Vorspalte-Labels) Spalte 1..99 (C1...99) geprüft; ggf. im Verhältnis zu einem konstanten Wert [+ , - , * , /].
and , or	value >= 10 and value <= 15;\$191966	Logische Operationen können mit ,and / or' kombiniert werden.
max min	max;\$00FF00 min;\$FF0000	Der Maximal-/ Minimalwert der Zahlenwerte wird mit der gewünschten Farbe hinterlegt.
maxNext<Zahl> minNext<Zahl>	maxNext2;\$00FF00 minNext3;\$FF0000	Der nächste (Next2) oder zweitmächste (Next3) usw. Wert unter dem Maximalwert bzw. über dem Minimalwert wird ermittelt.
max+ - * /<Zahl> min+ - * /<Zahl>	max*2;\$FF0000 min+10;\$FF0000	Der Zellenwert wird im Verhältnis zum Maximal-/ Minimalwert geprüft und bei Zutreffen markiert.
max+ - * /<Zahl>% min+ - * /<Zahl>%	max-10%;\$FF0000 min+50%;\$FF0000	Der Zellenwert wird im prozentualen Verhältnis zum Maximal-/ Minimalwert geprüft und bei Zutreffen markiert.

7.12 CellFontColor

Syntaxstruktur:


`CellFontColor=<Bedingung>;<Farbangabe>`

Analog zu CellColor können mit CellFontColor die Schriftfarben in Tabellenzellen in Abhängigkeit von der Zellenbesetzung beeinflusst werden.

Die Formulierung der Zahlenwerte-Bedingungen entspricht den in Tabelle 7.8 dargestellten Regeln.

Mit `POWERCHARTOPTION "CellFontColor=value>=25;$FF0000"` werden beispielsweise alle Werte ab 25 in der Tabelle mit roter Schriftfarbe markiert:

POWERCHARTOPTIONS



CellFontcolor

	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
sehr gut	7	14	12	18	4
gut	24	19	29	28	36
teils/teils	24	34	26	25	18
schlecht	21	23	20	21	36
sehr schlecht	24	10	14	9	7

Dummy-Daten


 GESS
GESS ChartingFactory
11

Abb. 7.15: POWERCHARTOPTION 'Cellfontcolor'

7.13 CellIcon

Syntaxstruktur:

`CellIcon=<Bedingung>;pos=L|C|R,resize=X|Y|B, arrange=B|F,<Bild>;`

Mit `CellIcon` können in Abhängigkeit von Zahlenwerten Bilder in Tabellenzellen eingefügt werden.

Für die Formulierung der Zahlenwerte-Bedingungen, siehe Tabelle 7.8.

Für die Positionierung (**pos**), Größeneinstellung (**resize**) und Anordnung (**arrange**) des Bildes stehen folgende Parameter zur Verfügung:

Tab. 7.9: Optionen für POWERCHARTOPTION 'Cellicon'

pos ={L C R}	L = Left C = Center R = Right
resize ={X Y B}	X = Zellenbreite Y = Zellenhöhe B = Both (Breite + Höhe)
arrange ={B F}	B = Bild in den Hintergrund (Background) F = Bild in den Vordergrund (Foreground)

Die folgende Syntaxsequenz bewirkt zum Beispiel, dass in der unten stehenden Tabelle Werte ab 30 durch ein Bild namens 'PIC_SMILE_GOOD' ersetzt werden:

```
POWERCHARTOPTION "CellIcon=value>=30;pos=C,resize=Y, arrange=F, PIC_SMILE_good"
```

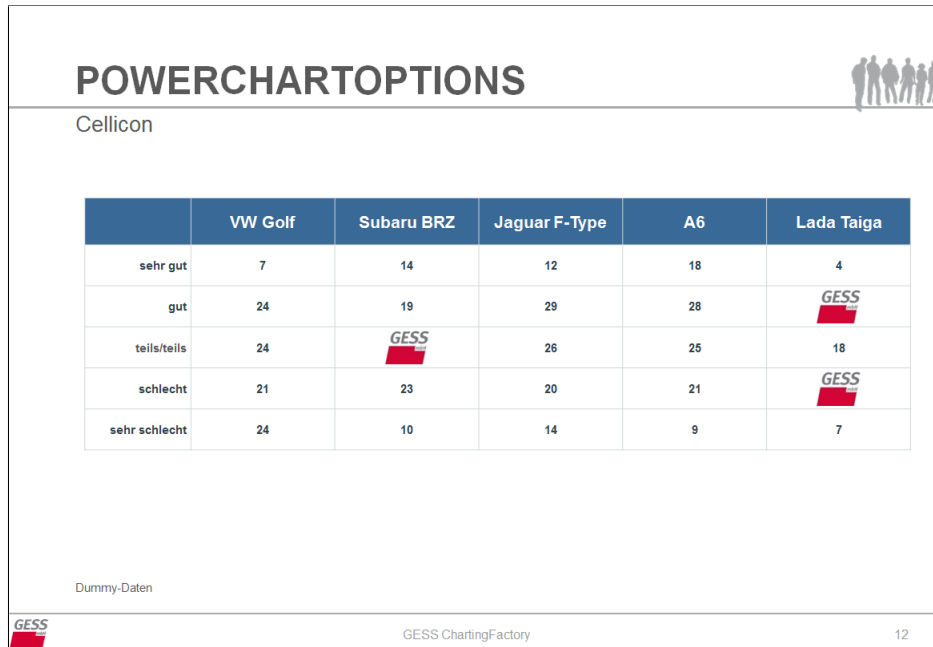


Abb. 7.16: POWERCHARTOPTION 'Cellicon'

7.14 PerRow/ PerCol

Mit PerRow/ PerCol können die oben aufgeführten Optionen zu CellColor und CellIcon auf Wunsch für jede Zeile (PerRow) oder Spalte (PerCol) ausgeführt werden.

Mit `POWERCHARTOPTION "CellColor=max;$FFFF00|PerRow"` wird hier etwa der höchste Wert je Zeile farblich markiert.

The screenshot shows a chart titled "POWERCHARTOPTIONS" with a subtitle "PerCol/Row". It displays a table with the following data:

	VW Golf	Subaru BRZ	Jaguar F-Type	A6	Lada Taiga
sehr gut	7	14	12	18	4
gut	24	19	29	28	36
teils/teils	24	34	26	25	18
schlecht	21	23	20	21	36
sehr schlecht	24	10	14	9	7

At the bottom of the chart, there is a "Dummy-Daten" label, a GESS logo, "GESS ChartingFactory", and the number "12".

Abb. 7.17: POWERCHARTOPTION 'Cellcolor|PerRow'

Während mit den POWERCHARTOPTIONs `CellColor`, `CellIcon` sowie dem Zusatz `Per Row/PerCol` einzelne Zellen(-inhalte) in Tabellen angesprochen werden, steuern die folgenden `Series`-Befehle Datenreihen in Diagrammen an.

7.15 SeriesColor

Syntaxstruktur:

`SeriesColor=<Bedingung>;<Farbangabe>`

Analog zu `CellColor` können mit `SeriesColor` Datenreihen, die in Diagrammen beispielsweise als Balken oder Linien dargestellt sind, in Abhängigkeit von ihrem Zahlenwert eingefärbt werden.

Für die Definition der Zahlenwerte/ -bereiche können die unter [Formulierung von Bedingungen für POWERCHARTOPTION 'CellColor'](#) erklärten Operationen genutzt werden, gleiches gilt für die in `CellColor` beschriebenen Möglichkeiten zur Farbangabe.

Auf dem folgenden Chartbeispiel wurden alle abgebildeten Datenwerte ab 30 in Signalgelb eingefärbt: `POWERCHARTOPTION "SeriesColor=value>=30;$FFFF00"`.

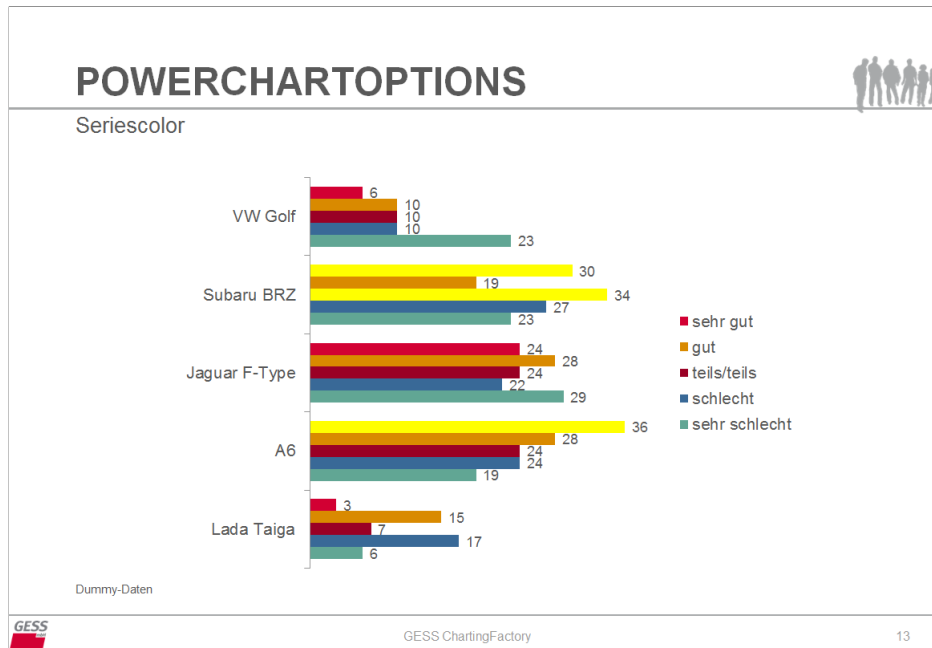


Abb. 7.18: POWERCHARTOPTION 'Seriescolor'

7.16 SeriesLabelColor

Syntaxstruktur:

`SeriesLabelColor=<Bedingung>;<Farbangabe>`

Mit `SeriesLabelColor` kann die Schriftfarbe der in Diagramme integrierten Labeltexte je nach Wertausprägung der entsprechenden Datenreihe verändert werden.

Die Bedingungen und Farbangaben erfolgen nach den in Tabelle 7.8 beschriebenen Optionen.

Hier werden zum Beispiel die Datenbeschriftungen für Werte unter 11 in Rot eingefärbt:
 POWERCHARTOPTION "SeriesLabelColor=value<=10;\$FF0000" .

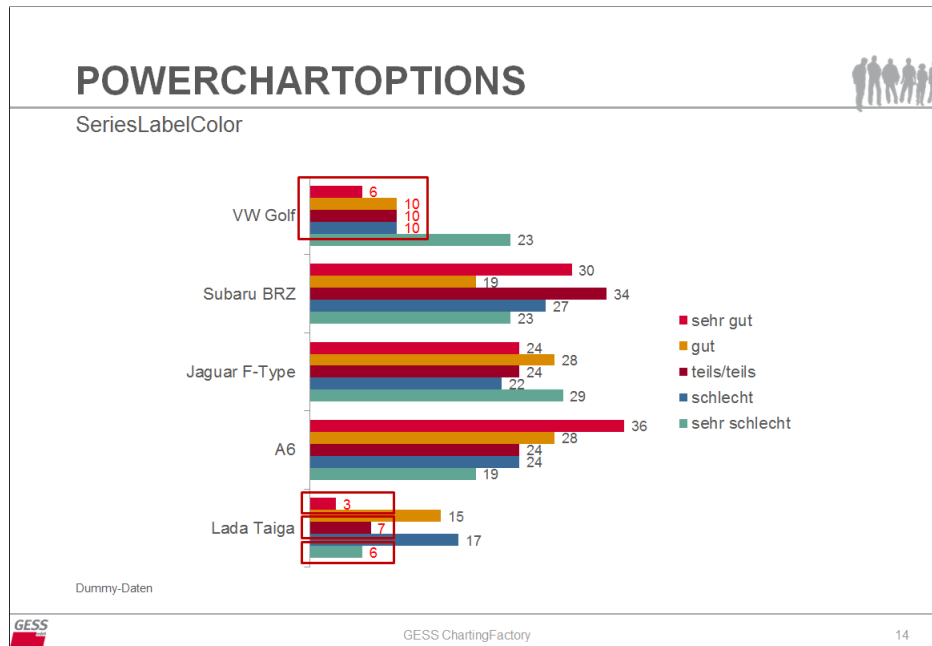


Abb. 7.19: POWERCHARTOPTION 'SeriesLabelcolor'

7.17 SeriesColorMarkstring

Syntaxstruktur:

SeriesColorMarkstring=<Text>;<Farbangabe>{;<Zahlenwert>}

Mit SeriesColorMarkstring können die Datenreihen (Balken/Linien etc.) von Diagrammen farbig markiert werden, wenn dessen Labels den im Statement definierten Text(-teil) enthalten.

Ist der exakte Labeltext nicht bekannt oder sollen mehrere Texte mit identischen Textteilen erkannt werden, können [Wildcards in Suchtexten](#) verwendet werden.

Die gewünschte Einfärbung kann über die Hexidezimal-Definition mit \$1122AA/ #1122AA oder mit einem RGB-Code 255,255,255 angegeben werden.

So werden im folgenden Beispiel alle Diagrammteile, die 'VW' enthalten (in diesem Fall ist dies die komplette Datenreihe namens 'VW Golf') in der Farbe '\$FFFF00' markiert:

```
POWERCHARTOPTION "SeriesColorMarkstring=VW*;$FFFF00"
```

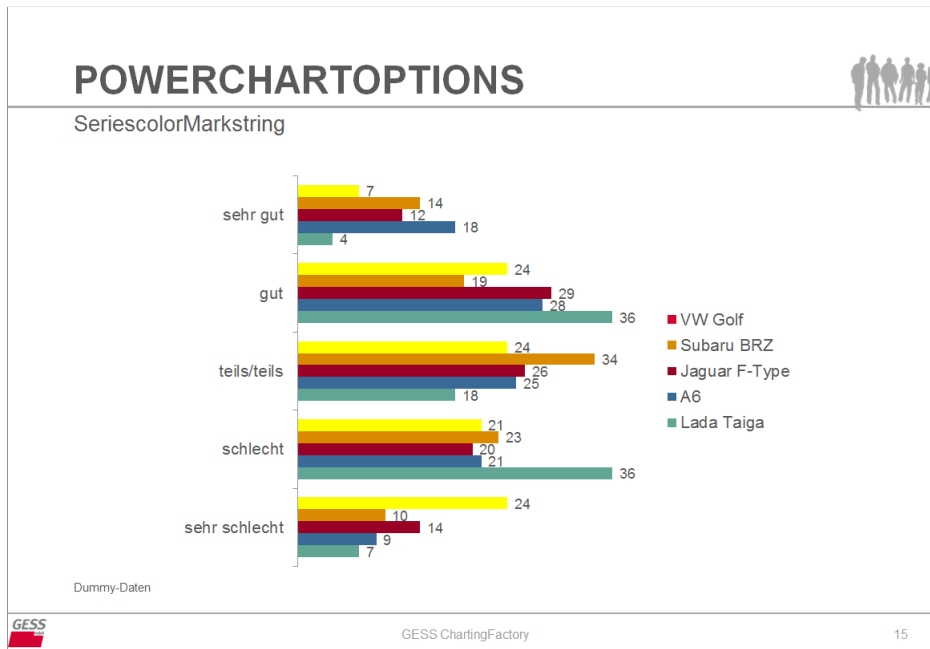


Abb. 7.20: POWERCHARTOPTION 'SeriesColorMarkstring'

In gestapelten oder geschichteten Diagrammen können außerdem einzelne Datenreihenteile angesprochen und mit unterschiedlichen Farben markiert werden. Unten werden etwa durch die Ergänzung ;1 die erste Datenreihe der Kategorie 'sehr gut' (VW Golf) in Gelb und die zweite Datenreihe (;2 für Subaru BRZ) in Türkis eingefärbt.

```
POWERCHARTOPTION "SeriesColorMarkstring=sehr gut;$FFFF00;1|
SeriesColorMarkstring=sehr gut;$00FFFF;2"
```

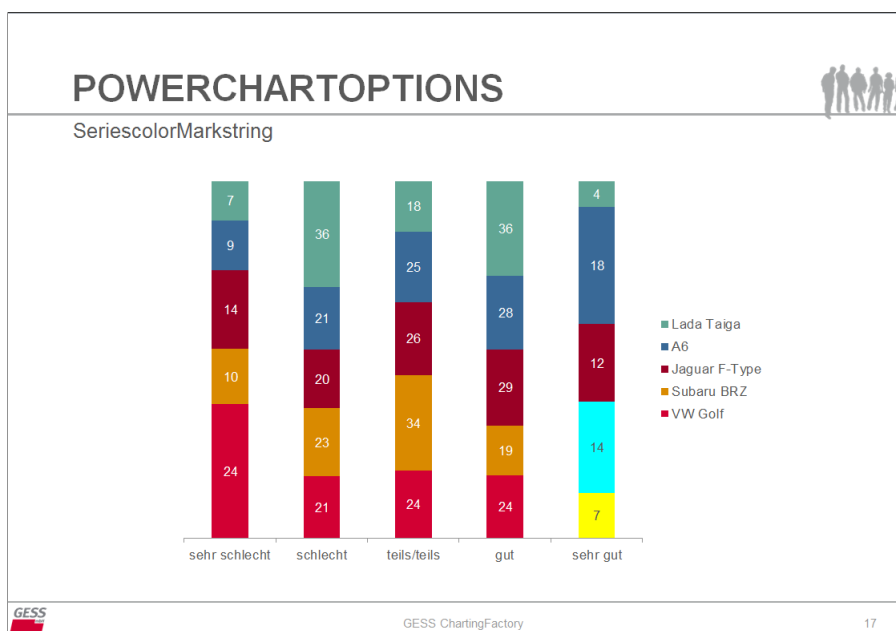


Abb. 7.21: POWERCHARTOPTION 'SeriescolorMarkstring'

7.18 SeriesLabelAdd

Syntaxstruktur:

SeriesLabelAdd = [<Ziffer>|<Buchstabe>]

Mit SeriesLabelAdd können Datenreihen-Beschriftungen in Diagrammen um zusätzliche Texte (zum Beispiel Signifikanzbuchstaben) ergänzt werden.

Bei Angabe einer Zahl wird der Inhalt der x-ten Zeile des Excel-Datenblattes als zusätzlicher Text an die Datenbeschriftung der Datenreihe angefügt. Bei Angabe eines Buchstaben wird der Inhalt der entsprechenden Spalte als zusätzlicher Text an die Datenreihen-Beschriftung gefügt.

Voraussetzung ist, dass der zu ergänzende Inhalt per POSITION zuvor in das Excel-Datenblatt in PowerPoint übertragen wurde.

Im unten stehenden Beispiel werden den Datenreihen-Beschriftungen Inhalte aus Zeile 15 hinzugefügt: POWERCHARTOPTION "SeriesLabelAdd=15" .

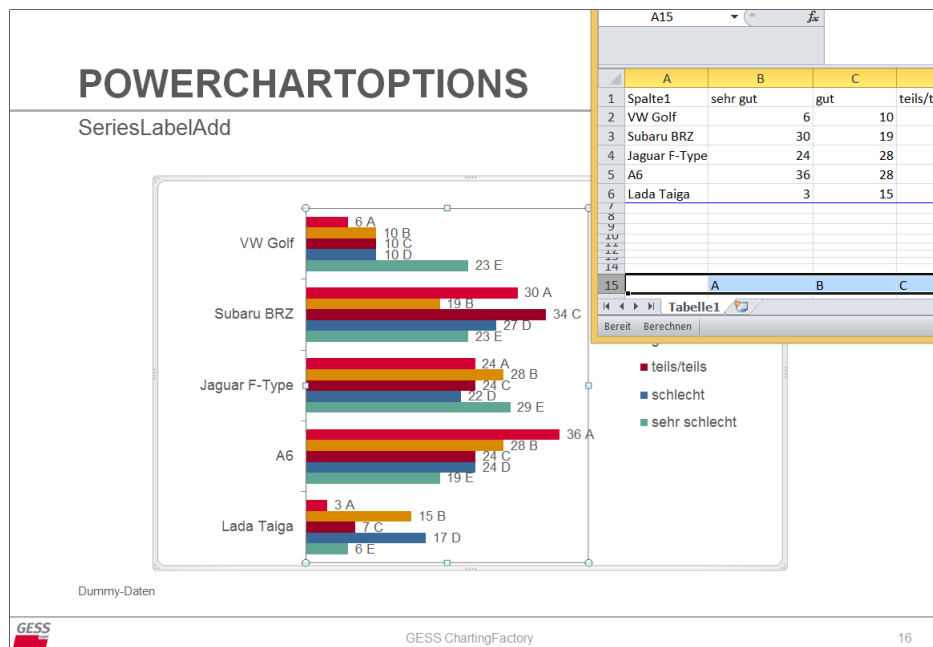


Abb. 7.22: POWERCHARTOPTION 'SeriesLabelAdd'

7.19 SeriesLabelReplace

Syntaxstruktur:

SeriesLabelReplace = [<Ziffer>|<Buchstabe>]

Mit SeriesLabelReplace können Datenbeschriftungen innerhalb eines Diagramms durch die Daten aus der angegebenen Zeile (Zahl) oder Spalte (Buchstabe) des Excel-Datenblattes ersetzt werden. So können beispielsweise Zahlenwerte durch Signifikanzbuchstaben ausgetauscht werden.

Analog zu SeriesLabelAdd (siehe 7.18) wird bei Angabe einer Zahl der Inhalt der entsprechenden Zeile angesprochen, bei Angabe eines Buchstaben werden die Zahlenwerte der Datenreihen durch den Inhalt aus der x-ten Spalte ersetzt.

Mit POWERCHARTOPTION "SeriesLabelReplace=I" wird etwa der Inhalt aus Spalte I anstelle der Zahlenwerte angezeigt:

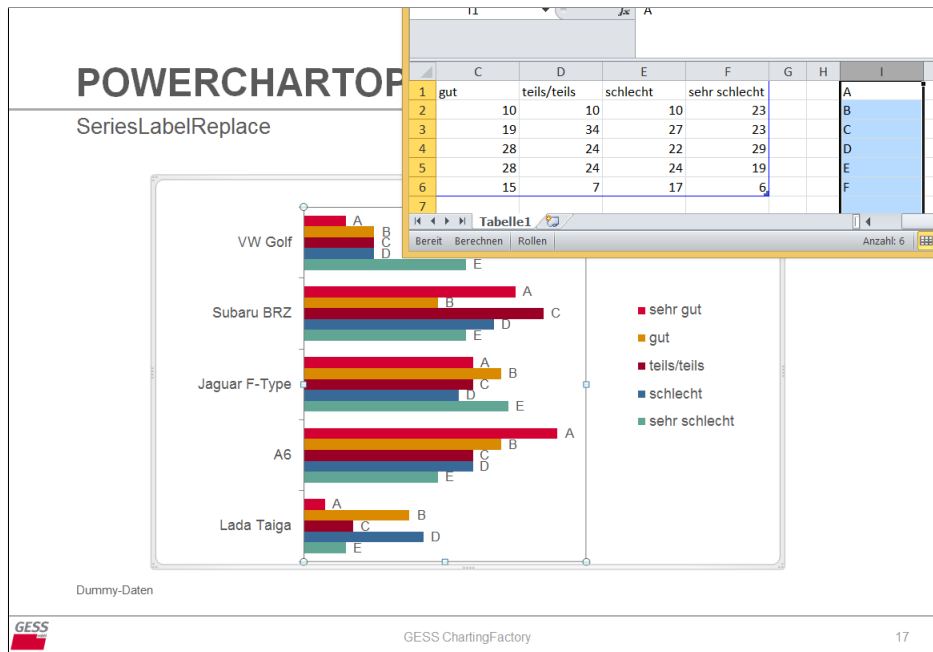


Abb. 7.23: POWERCHARTOPTION 'SeriesLabelReplace'

7.20 FillColorObject

Syntaxstruktur:

FillColorObject=<PP-Element>;<Bedingung>;color=<Farbangabe>{;cell=[<row>,<column>]}

Mit FillColorObject können schließlich ganze PowerPoint-Objekte in Abhängigkeit von einer definierten Bedingung eingefärbt werden. Geprüft wird der Inhalt eines Textfeldes (Texte und Zahlenwerte) oder einer Tabellenzelle (Zahlenwerte). Trifft die gestellte Bedingung zu, wird das betreffende PowerPoint-Element (Grafik, Tabelle oder Bild) im definierten Farbton eingefärbt.

Bedingungen können anhand folgender Operatoren formuliert: <, >, <=, >=, <>, = sowie mit and/or verknüpft werden.

Im ersten Beispiel dient die Textbox namens TEXT_05 als Prüffeld. Übersteigt die darin enthaltene Basis 1000, soll der Diagramm-Hintergrund von GRAPH_01 hellgrau eingefärbt werden:

```
FILL TEXT_05 WITH CONCAT (PHYSICALRECORDS FORMAT "#")
POWERCHARTOPTION "FillColorObject=GRAPH_01;TEXT_05 >= 1000;color=#E6E6E6"
```

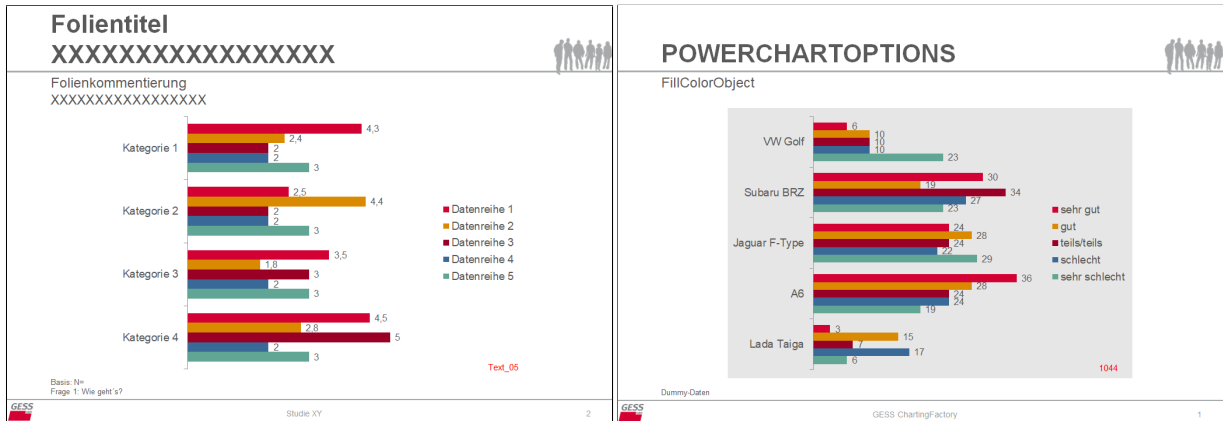


Abb. 7.24: Vorlage 1 für POWERCHARTOPTI-Abb. 7.25: POWERCHARTOPTION 'FillColorObject' 'FillColorObject' 1

In Beispiel 2 wird der Inhalt einer Tabellenzelle geprüft (hier: Reihe 4, Spalte 1 in TABLE_01). Liegt der Zahlenwert zwischen 35 und 50, erfolgt eine Grünfärbung der PowerPoint-Form namens Pfeil.

```
POWERCHARTOPTION "FillColorObject=Pfeil;TABLE_01>=35 and TABLE_01<=50;
color=0,153,0;cell=4,1"
```

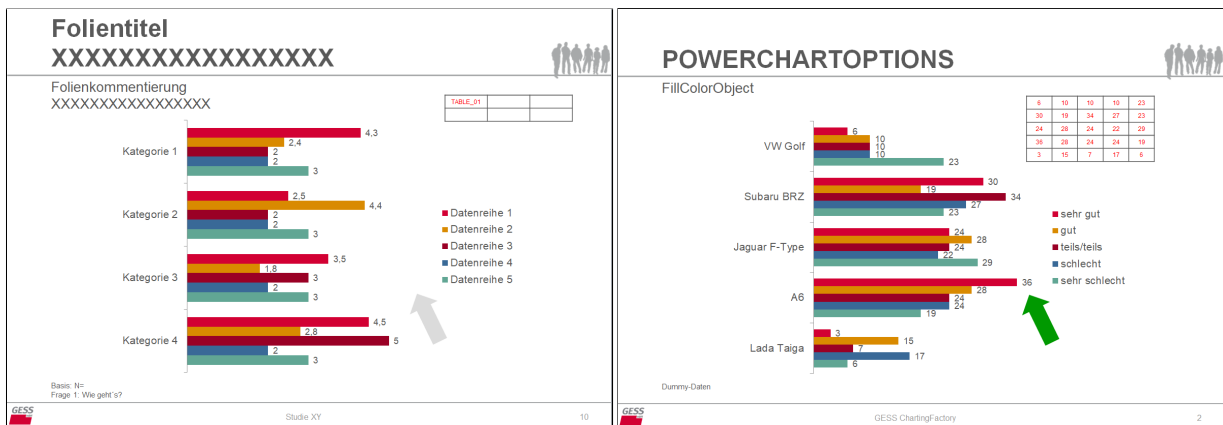


Abb. 7.26: Vorlage 2 für POWERCHARTOPTI-Abb. 7.27: POWERCHARTOPTION 'FillColorObject' 'FillColorObject' 2

7.21 DeleteObject

Syntaxstruktur:

```
DeleteObject=<PP-Element>{;<TEXT_[xx] Bedingung>{;<NDC>}}
```

Mit DeleteObject können PowerPoint-Objekte, die auf der verwendeten PowerPoint-Vorlage enthalten sind, jedoch für die aktuelle Ausgabe nicht benötigt werden, gelöscht werden. Dies hat den Vorteil, dass auf allzu viele ähnliche Chartvorlagen verzichtet werden kann.

Die links stehende Vorlage enthält beispielsweise eine Tabelle (TABLE_01), die für die rechts stehende Ausgabe nicht benötigt wird. Mit `POWERCHARTOPTION "DeleteObject=TABLE_01"` kann diese gelöscht (und in diesem Fall beispielsweise der Platz via `LabelIconReplace` mit einem Bild im Textfeld TEXT_05 ausgefüllt) werden.

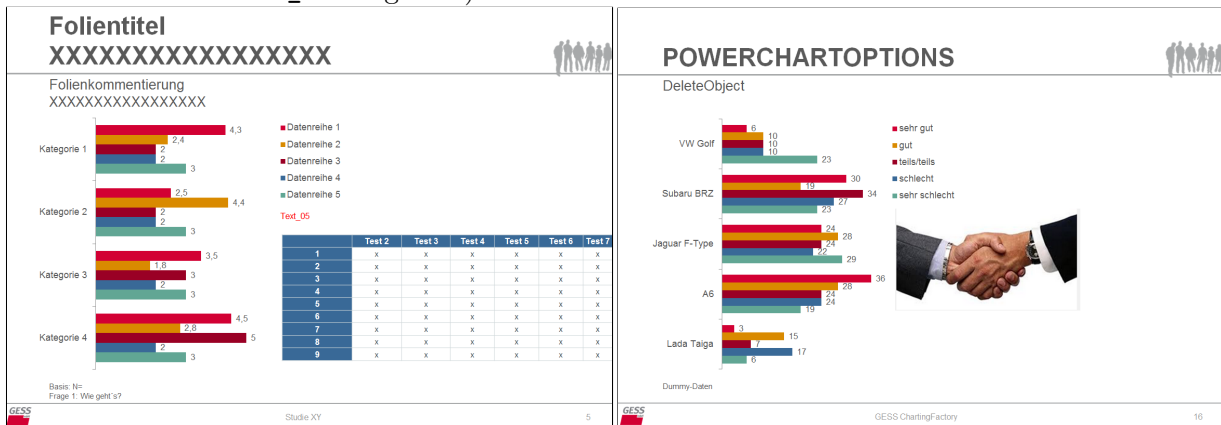


Abb. 7.28: Vorlage für POWERCHARTOPTI-Abb. 7.29: POWERCHARTOPTION 'DeleteObject' 'DeleteObject'

Ergänzend kann das Löschen von Elementen an eine Bedingung geknüpft werden: Geprüft wird der Inhalt einer Textbox (Texte und Zahlenwerte). Trifft die formulierte Bedingung zu, wird das betreffende PowerPoint-Element gelöscht. Standardmäßig wird die überprüfte Textbox ebenfalls gelöscht, dies kann mit der Zugabe ;NDC unterdrückt werden.

Bedingungen können analog zu `FillColorObject` formuliert und verknüpft werden.

Beispiel:

```
FILL TEXT_01 WITH CONCAT (PHYSICALRECORDS FORMAT "#")
FILL TEXT_02 WITH CONCAT ("Freier Text")
POWERCHARTOPTION "DeleteObject=GRAPH_01;TEXT_01 <= 200 or TEXT_01 > 300|
DeleteObject=TABLE_01;TEXT_02 = *Text;NDC"
```

Trifft die gestellte Bedingung nicht zu, wird keines der PowerPoint-Elemente gelöscht.

8 Daten-Aktualisierungen

Mit dem Statement zum POWERCHARTREIMPORT sind mehrere Befragungsgruppen, -regionen und -wellen, kurzfristige Datenänderungen, -ergänzungen sowie stetige Datenaktualisierungen problemlos handhabbar. Die in das Ergebnischart eingefügten Daten werden aktualisiert und kleine Layoutänderungen, Umsortierungen in der Chart-Reihenfolge und andere Veränderungen, die nachträglich und händisch in PowerPoint vorgenommen wurden, bleiben bestehen und werden nicht überschrieben.

Wurden beispielsweise nachträglich Bearbeitungen auf dem Ausgabe-Chart vorgenommen, wie auf diesem Chart, dem zwei zusätzliche Kommentar-Elemente hinzugefügt wurden

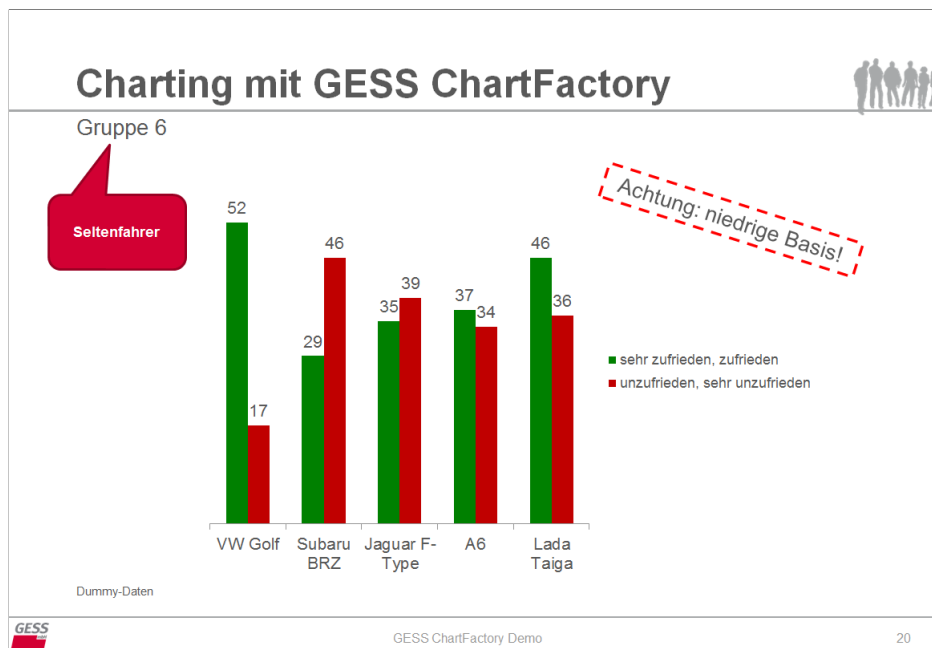


Abb. 8.1: Manuell ergänztes Chart

..., können diese beim Einspielen neuer Daten beibehalten werden. Dies gelingt, indem die Datengrundlage im Skript angepasst (z.B. `SPSSINFILE = "Daten_neu.sav";`) und vor der ersten Tabellierungs- und POWERCHART-Anweisung folgendes Statement eingefügt wird: `POWERCHARTREIMPORT = YES;`

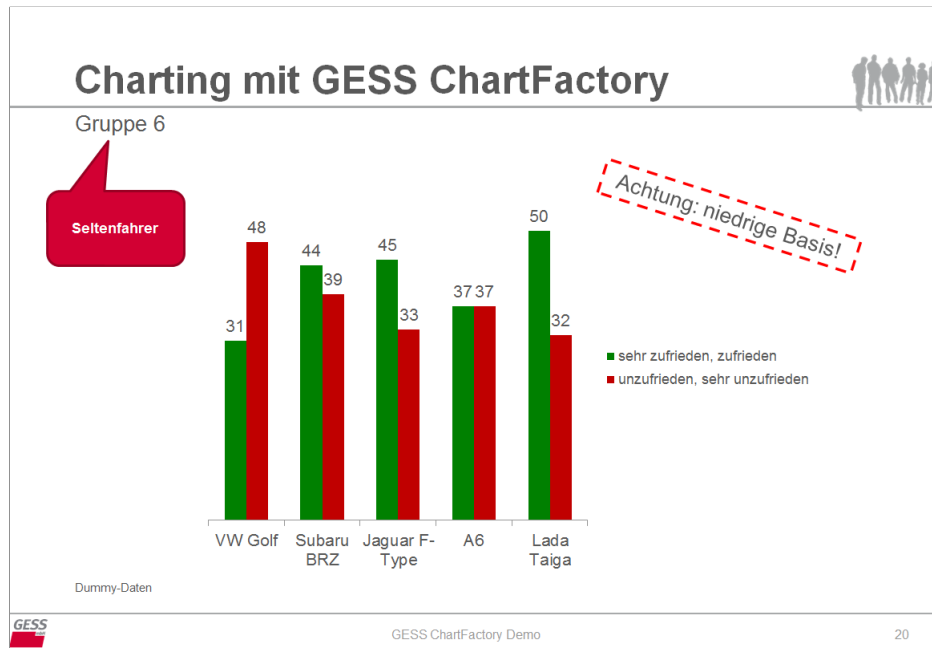


Abb. 8.2: Mit 'POWERCHARTREIMPORT' aktualisiertes Chart

Wie man sieht, wurden die neuen Daten eingespielt und die nachträglich in der Ausgabedatei gemachten Änderungen übernommen.

Was dabei passiert, ist Folgendes:

Die Software lässt nicht das gesamte Skript erneut durchlaufen — denn dann wäre die ursprüngliche Vorlage-Datei die Grundlage unseres Charting-Prozesses und alle händischen Bearbeitungen der Ausgabe-Datei verloren. Stattdessen nimmt sich die Software die ursprüngliche (nachträglich bearbeitete) Ausgabe-Datei als (neues) PPTEMPLATE, legt eine Kopie dessen an, benennt diese mit 'Dateiname_Copy_Before_Reimport_Datum_Uhrzeit' und erstellt eine neue Ausgabedatei, in die die neuen Daten eingesetzt werden.

Anschließend liegen drei PowerPoint-Dateien vor, im Beispiel-Fall:

1. PP-Vorlage.pptx — Vorlagedatei (PPTEMPLATES)
2. PP-Ausgabe_Copy_Before_Reimport_20151014_135153.pptx — Ausgabedatei, wie sie laut Skript erstellt wird
3. PP-Ausgabe.pptx — neue Ausgabedatei mit aktualisierten Daten

9 GESStabs-spezifische Hinweise

Für die Nutzung von GESStabs ist ein separates Handbuch bereitgestellt, daher wird im Rahmen dieser Dokumentation nicht vertiefend auf die Tabellierung mit GESStabs eingegangen. Einige Funktionalitäten sind jedoch im Zusammenhang mit der GESS ChartFactory beachtenswert und werden im Folgenden erläutert.

9.1 Tabelleninhalte

Grundsätzlich können nur Inhalte in PowerPoint übertragen und abgebildet werden, die in der referenzierten GESStabs-Tabelle enthalten sind. Enthält eine GESStabs-Tabelle nur vier Zeilen, werden im POSITION-Statement aber beispielsweise sieben Zeilen angegeben (`ROWS 1:7`), werden nur die vier vorhandenen Zeilen übertragen. Wird der `Vartitle` der Variable Q1 in der GESStabs-Tabelle nicht angeführt, bleibt das Textfeld, das mit `FILL TEXT_01 WITH CONCAT (VARTITLE V1)` befüllt werden soll, leer.

9.2 Leere Tabellenzellen

Mit der GESStabs-Einstellung `TABLEFORMAT = -NOZERODASH;` wird für leere Tabellenzellen Folgendes eingestellt: In den Tabellen werden Zellen, die keine Daten enthalten, mit einem Minus gefüllt. Bei der Datenübertragung nach PowerPoint werden diese grundsätzlich als 'Nullen' behandelt, sodass ein reibungsloser Ablauf — etwa bei der Berechnung neuer aus bestehenden Datenreihen mittels `POWERCHARTOPTION Formula` (siehe 7.8) — sichergestellt ist. Dargestellt werden leere Tabellenzellen in PowerPoint-Tabellen als Minus, in Datentabellen von Diagrammen als Nullen.

9.3 Mehrseitige Tabellen

Werden Datentabellen zum Druck in mehrere Teiltabellen zerlegt und auf mehreren Seiten dargestellt¹, hat dies Konsequenzen für die Datenübertragung mittels `POWERCHART`.

Die Charts werden auf Basis der ersten Seite einer (zerlegten) Tabelle erzeugt. Werden im `POWERCHART`-Statement Datenzellen angesprochen, die nicht in der ersten Teiltabelle enthalten sind, werden diese als leere Zellen behandelt und entsprechend nicht dargestellt. Im Message-File und Compiler-Protokoll werden entsprechende Warnungen ausgegeben.

Wir empfehlen daher, im Zuge des Chartings von langen Tabellen keine Seitenaufteilungen in GESStabs vorzunehmen.

¹Siehe GESStabs-Optionen `SLICE`, `LSLICE`, `MAXCOLSPERTABLEPAGE`, `MAXROWSPERTABLEPAGE`.

9.4 Formatierung von Texten

In Tabellierungsbefehle können Steuerinformationen zur Textformatierung integriert werden, die von GESS ChartFactory erkannt und verarbeitet werden.

```
LABELS kopf1 =
1 "<FONT BOLD SIZE=16>VW Golf </FONT>"
2 "<FONT UNDERLINE>Subaru BRZ </FONT>"
;
```

Diese werden jedoch in die Texte selbst eingetragen und erscheinen so in den Datentabellen, z.B.:

Col %	sehr gut	gut
Base	132	272
Modellvariante		
VW Golf 	6 %	10 %
Subaru BRZ	30 %	19 %

Abb. 9.1: 'FONT' in Tabellentext

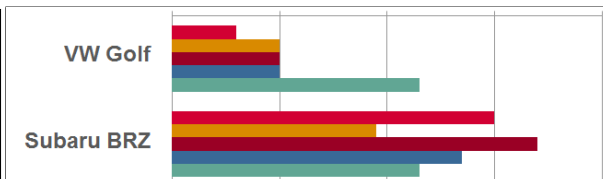


Abb. 9.2: 'FONT' im Diagramm

Mit dem POWERCHARTTEXTEXTRA-Statement können nun generell gültige Formateinstellungen für Labeltexte an das Charting weitergegeben werden, die in den anderen Ausgabemedien (PS, PDF, HTML, Excel usw.) aber nicht erscheinen. Die entsprechende Syntaxstruktur lautet:

```
POWERCHARTTEXTEXTRA <[LABELS X | LABELS Y]> =
```

```
<"[BOLD | UNDERLINE | SIZE = Nummer]">;
```

Für das obenstehende Beispiel entsprechend:

```
POWERCHARTTEXTEXTRA LABELS X = "BOLD SIZE = 16";
```

9.5 Multiuser

Die Datei 'POWERCHART.xlsb', die PowerPoint und den Charting-Vorgang startet, muss während des Programmlaufs im exklusiven Schreibzugriff sein. Eine Nutzung durch mehrere Anwender ist aufgrund interner Abläufe aber nicht immer vermeidbar. Kollisionen oder Programmabstürze bei mehreren simultanen Nutzern können mit POWERCHARTMULTIUSER = [YES | NO]; vermieden werden. Bei Aktivierung der Multiuser-Funktion wird eine benutzerspezifische Kopie von POWERCHART.xlsb erzeugt und gestartet.

9.6 Berechnete Tabellenelemente

Ergebnisse von im Tabellierungsprozess berechneten Zellen (umsetzbar z.B. mittels COMPUTE oder CALCULATECOLUMN) werden in History-kompatiblen Zellen gespeichert, sind für POWERCHART sichtbar und werden wie übliche Tabellenelemente behandelt. Sie können daher problemlos für den Datenimport in PowerPoint verwendet werden.

10 PowerPoint-spezifische Hinweise

Die Voraussetzung für den Einsatz der GESS ChartFactory ist die Vorbereitung von Vorlagecharts in MS PowerPoint (siehe dazu [Vorbereitung der PowerPoint-Vorlage](#)). Diesbezüglich werden im Folgenden einige Hinweise gegeben, die für bestimmte GESS ChartFactory-Funktionen nützlich sind.

10.1 Kontrolle von Formatierungen

Empfohlen wird, Diagramme und Tabellen in der Vorlage-Datei stets so anzulegen, dass sie die maximale Anzahl an Datenreihen und -zellen enthalten. So kann die Formatierung aller Farben und Texte vorbereitet und kontrolliert werden.

Wird ein PowerPoint-Element im Zuge des Autocharting mit mehr als der im Template definierten Anzahl an Datenreihen bzw. -zellen befüllt, kann es zu nicht erwünschten Formatierungen, etwa bei der Einfärbung von Balken, kommen.

10.2 Diagramm-interne Beschriftungen

GESS ChartFactory kann grundsätzlich nicht auf Beschriftungen zugreifen, die in PowerPoint-Diagramme integriert sind¹. Eine Ausnahme bilden global gültige Formateinstellungen für Labeltexte via POWERCHARTTEXTEXTRA (siehe [Formatierung von Texten](#)).

Um einzelne Texte im automatisierten Charting-Prozess ansprechen und individuell formatieren oder beispielsweise durch Bilder ersetzen zu können², müssen diese in Textboxen oder als Tabelle ins Chart eingefügt werden. Für eine Anleitung zur konkreten Umsetzung, siehe Kapitel [4.2](#) und [4.3](#).

10.3 Skalierung einzufügender Bilder

Mit POWERCHARTOPTION `LabelIconReplace` können Texte innerhalb von Tabellen und Textboxen durch Bilder ersetzt werden (siehe [LabelIconReplace](#) und für eine konkrete Anwendung auch [Tabellen](#)). Um die einzufügenden Bilder im Zuge dessen skalieren zu können (`RESIZE=X|Y|B`), muss die Bildeinstellung 'Seitenverhältnis sperren' im Grafik-Menü deaktiviert sein.

¹Siehe hierzu auch Kapitel [4.1](#).

²via POWERCHARTOPTION [LabelIconReplace](#)

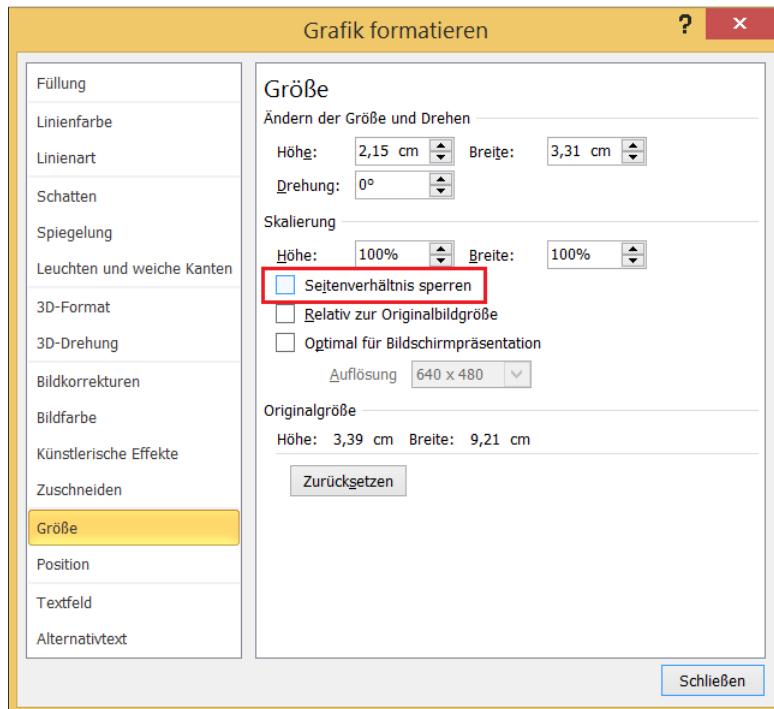


Abb. 10.1: Deaktivieren von 'Seitenverhältnis sperren'

10.4 Ausblenden von Nullen in Diagrammen

Wie in Kapitel 9.2 beschrieben, werden leere Tabellenzellen als Nullen in die Datentabellen von Diagrammen übertragen (mit der entsprechenden Einstellung). Sollen diese Nullen grundsätzlich nicht in der Diagramm-Ansicht abgebildet werden³, muss in der PowerPoint-Vorlage für das betreffende Diagramm unter 'Datenbeschriftung formatieren' die folgende markierte Formatierung für Zahlen ausgewählt werden.

³Dies ist zu empfehlen, da sonst Beschriftungen eingeblendet werden, zu denen es aufgrund der fehlenden Zellenbesetzung keine Datenreihe gibt.

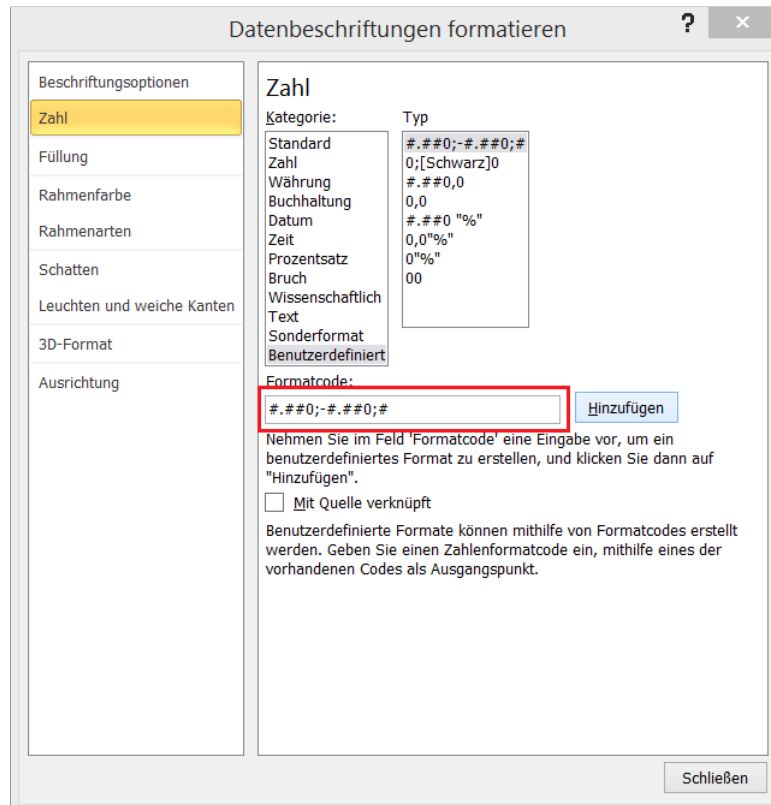


Abb. 10.2: Ausblenden von Nullen

10.5 'AutoWiederherstellen'-Funktion

Wie unter [Technische Voraussetzungen](#) erklärt, deaktiviert die aktuelle Software-Version der GESS ChartFactory die 'AutoWiederherstellen'-Funktion der Microsoft-Produkte. So wird ein stabilerer und schnellerer Charting-Ablauf erreicht.

Wenn gewünscht, kann diese Funktion in den Excel-/ PowerPoint-Optionen aber wieder aktiviert werden, unter 'Speichern → AutoWiederherstellen-Info: alle x Minuten speichern'.

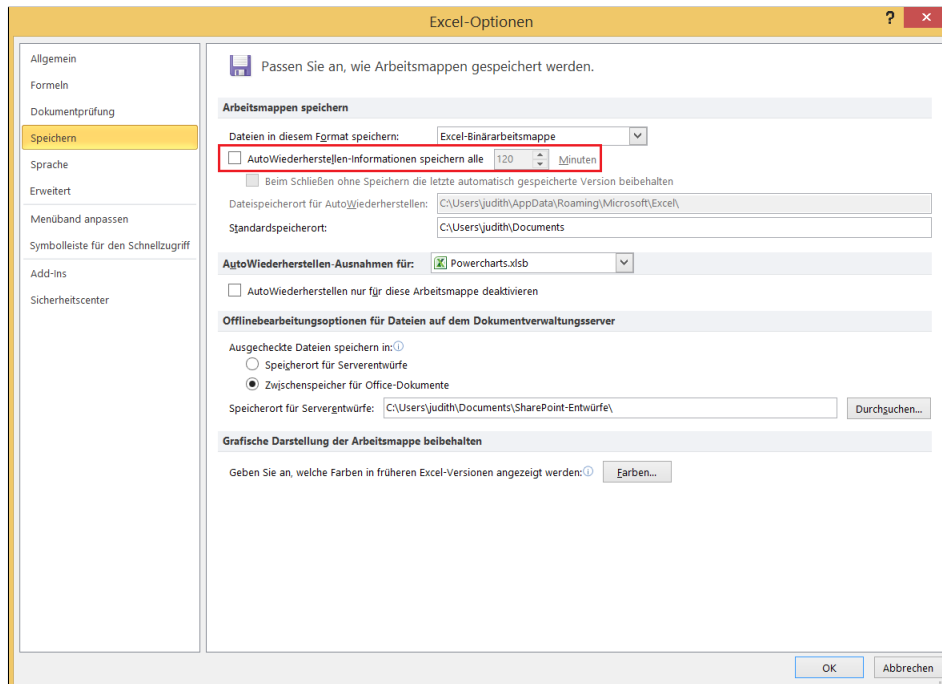


Abb. 10.3: 'AutoWiederherstellen'-Funktion

11 Syntax

Folgend wird die Skriptsprache der GESS ChartFactory und der Aufbau eines POWERCHART-Statements, einschließlich aller Optionen, erklärt.

Zunächst wird ein TABLE-Statement nach GESStabs-Konventionen formuliert, z.B. TABLE = a by b;

11.1 Basisbefehl

Dann folgt ein POWERCHART-Statement, das folgende **Grundstruktur** aufweist:

```
POWERCHART
KEY <string>
.....
TEMPLATE BY [ROWS | COLUMNS] <{<count> : <Templatenr.>}*n [ELSE <Templatenr.>]>
oder
TEMPLATE [<Nummer> | p]
.....
GRATAB <PP-Name>
.....
POSITION = | ROWS <Nummer> COLUMNS <Nummer>
oder
| = ROWS <Labelcode> COLUMNS <Labelcode>
.....
;
```

Beispiel:

```
POWERCHART
KEY Tab_001
TEMPLATE BY ROWS <2:7 3:8 ELSE 9>
GRATAB GRAPH_01
| = ROWS 65002 65003 COLUMNS 1:6
;
```

Oder auch:

```
POWERCHART
KEY Tab_001
TEMPLATE 1
GRATAB GRAPH_01
POSITION = | ROWS 1:3 COLUMNS 3 4
;
```

Zu den einzelnen Bestandteilen:

POWERCHART markiert den Start der Charting-Anweisung.

KEY ist eine eindeutige Kennung für die spezifische POWERCHART-Anweisung.

Jedes POWERCHART-Statement muss mit einem eindeutigen KEY versehen sein. Dieser Schlüssel wird verdeckt in der PowerPoint-Präsentation gespeichert und dient der fehlerfreien Identifizierung des

spezifischen Vorgangs und als eindeutige Kennung für den Daten-Reimport zu einem späteren Zeitpunkt¹.

POWERCHART-Anweisungen ohne KEY oder mit einem KEY, der im selben GESStabs-Lauf bereits verwendet wurde, führen zu einem Programmabbruch mit Syntaxfehler-Meldung.

TEMPLATE steuert das gewünschte Vorlage-Chart aus der PPTEMPLATES-Datei an.

Voraussetzung für die Charterstellung ist eine PowerPoint-Datei, die als Vorlage für die zu generierenden Charts dient². Mit **TEMPLATE** <Seitennummer> bzw. **p** benennt man die entsprechende Seite aus der Vorlagedatei, die in die Ausgabedatei übertragen und mit Inhalten gefüllt werden soll.

Mit der einleitend zweitgenannten Variante können in Abhängigkeit vom Aussehen der GESStabs-Tabelle unterschiedliche Vorlage-Charts befüllt werden. Je nach Zeilen- bzw. Spaltenanzahl der zugrundeliegenden Tabelle wird das entsprechend definierte Vorlage-Chart angesprochen. Für **COUNT** sind Werte zwischen 1 und 20 zulässig.

GRATAB steuert das zu bearbeitende Element auf dem Vorlagechart an.

GRATAB ist ein Kunstwort aus 'GRAfik' und 'TABelle'. Mit jedem **POWERCHART**-Statement kann man sich auf genau eines dieser **GRATAB**-Elemente eines Vorlagecharts beziehen. Die Elemente müssen im Vorlagechart nach bestimmten Konventionen benannt werden³.

Sollen mehrere Tabellen/ Grafiken auf einem Chart angesprochen werden, werden hierfür mehrere **POWERCHART**-Statements formuliert, die ab dem zweiten Statement mit **TEMPLATE p** (p=previous) verkettet werden.

(POSITION) = | ROWS [...] COLUMNS [...] gibt die zu verwendenden Datenbereiche in der Tabelle an.

Für die Ansteuerung des im Element darzustellenden Tabellenbereiches gibt es grundsätzlich zwei Möglichkeiten:

1. **| = ROWS <Labelcode> COLUMNS <Labelcode>**

Mit dieser Syntaxvariante werden Labelcodes angesprochen.

Mit **| = ROWS 65002 65003 COLUMNS 1:6** werden also zum Beispiel die Codes 65002 und 65003 (übliche Bezeichnungsstruktur für Overcodes) der Zeilenvariable und die Codes 1 bis 6 der im Tabellenkopf stehenden Variablen ausgewählt.

Dies stößt allerdings an Grenzen, wenn z.B. eine GESStabs-Tabelle aus mehreren Variablen zusammengesetzt ist (dies ist u.a. bei Mittelwerte-Übersichten der Fall): Dann bestehen beispielsweise die Zeilen nicht aus Labelwerten, sondern aus Kennwerten verschiedener Variablen, die nicht anhand von Labelcodes adressiert werden können. Die einzelnen Elemente solcher Tabellen können dann als **ROWS X|Y/z** (X/Y für die Spalten-/ Zeilenvariable, z für den Labelcode), also beispielsweise mit **ROWS X 1**, **ROW Y 3** angesprochen werden, die in einem Statement miteinander verbunden werden, zum Beispiel:

```
=
| ROWS 1/1 COLUMNS 65002 65003
| ROWS 2/1 COLUMNS 65002 65003
;
```

¹Siehe Kapitel 8.

²Siehe Kapitel über [Vorbereitung der PowerPoint-Vorlage](#).

³Siehe für diese Regeln: [Vorbereitung der PowerPoint-Vorlage](#).

Eine Alternative bietet das Schlagwort `POSITION`:

2. `POSITION = | ROWS <Nummer> COLUMNS <Nummer>`

Damit werden die zu verwendenden Tabellenbereiche als Zeilen- und Spaltennummern direkt benannt, also z.B. Zeilen 1 bis 3 sowie Spalten 3 und 4 mit `POSITION = | ROWS 1:3 COLUMNS 3 4`.

Die Adressierung mittels `POSITION` ist gegenüber solchen Feinheiten wie der Unterscheidung zwischen Variablennummern und Labelcodes ignorant und ermöglicht daher auch ungeübten oder Nicht-Tabellierern einen einfachen und eindeutigen Verweis auf die Tabelleninhalte. Ebenso kann das `POSITION`-Schlagwort dem ganzen Befehl oder nur der Reihen-/ Spaltenangabe vorangestellt werden.

Auch können beide Varianten vermischt werden, typische syntaktische Formen sind:

```
= | ROWS 1:5 COLUMNS POSITION 1
= | ROWS POSITION 1:5 COLUMNS 65002
POSITION = | ROWS 1:5 COLUMNS 1
```

Ist (noch) nicht klar, wie viele Zeilen/ Spalten eine zu verchartende Tabelle enthält, kann mit `1:99` sichergestellt werden, dass alle vorhandenen Tabellenbereiche von `POWERCHART` aufgegriffen werden. Für die erste Lösung muss diesem ein `POSITION` vorangestellt werden, z.B. `| = ROWS POSITION 1:99 COLUMNS 65002 65003`, in der zweiten Variante wird dies wie übliche Zeilen-/ Spaltenangaben benutzt, etwa `POSITION = | ROWS 1:99 COLUMNS 3 4`.

Neben den reinen Datenbereichen einer Tabelle, die mit Ziffern adressiert werden (sei es als Labelcodes oder Tabellenbereiche), können auch spezielle Tabellenbereiche angesteuert werden und zwar mit folgenden Schlagworten:

<code>ROWS</code>	<code>COLUMNS</code>	
<code>LABELS</code>	<code>LABELS</code>	Labeltexte der in den Zeilen/ im Kopf stehenden Variablen
<code>ABSROW</code>	<code>ABSCOLUMN</code>	absolute (ggf. gewichtete) Häufigkeiten der in den Zeilen/ im Kopf stehenden Variablen
<code>PHYSICALROW</code>	<code>PHYSICALCOLUMN</code>	ungewichtete Häufigkeiten der in den Zeilen/ im Kopf stehenden Variablen
<code>TOTALROW</code>	<code>TOTALCOLUMN</code>	Zeilen-/ Spaltentotal

Beispiel: `POSITION = | ROWS LABELS COLUMNS 1:5`

11.2 Optionale Befehlsbestandteile

Optional können dem `POWERCHART`-Statement weitere Elemente hinzugefügt werden.

`FILL TEXT_[xx] WITH CONCAT (<Element>*n)` ist das Befehlsschlagwort für das Befüllen von Textfeldern⁴ mit folgenden möglichen Inhalten (Elementen):

⁴Ab GESStabs-Version 5.1.3.5. Um die Lauffähigkeit bisheriger Skripte zu gewährleisten, wird die vorherige Variante `FILL TEXT_[xx] WITH [FREETEXT "<Freier Text>" | GESStabs-Element]` weiterhin unterstützt, jedoch zukünftig nicht erweitert. Wir empfehlen den Einsatz der aktuellen Variante unter Verwendung von `CONCAT` – mit zwei Ausnahmen: 1. Formatierung freier Texte und 2. Ersatz einer Textbox durch eine Grafik via `LabelIconReplace`. In diesen beiden Fällen kommt unverändert `FILL TEXT_[xx] with FREETEXT` zum Einsatz, für Anwendungsbeispiele siehe Abschnitt 4.2.

“Freier Text“ ’Freier Text’	freier Text, wird auf dem Chart abgebildet
VARTEXT <Variablenname>	Vartext einer Variable
VARTITLE <Variablenname>	Vartitle einer Variable
VALUELABELS [x/y] <Labelnummer>	Text eines spezifischen Labelcodes
VALUELABELS [x/y] POSITION <Nummer>	Labeltext an ausgewählter Tabellenposition
TABLENAME	Tabellenname
TABLETITLE	Inhalt der Tabletitle-Box über der Tabelle
TABLEFILTER <Nummer>	Bezeichnung des Tabellenfilters
TOPTTEXT	Inhalt der oberen Textbox
BOTTOMTEXT	Inhalt der unteren Textbox
HEADER	Kopfzeile
FOOTER	Fußzeile
INSTITUTION	Name des Kunden (links unten an der Tabelle)
DOCUMENT	Dokumentenkenzeichnung (rechts unten an der Tabelle)
CONTENTKEY	Contentkey der Tabelle
GRAPHPROJECT	zusätzlicher Bezeichner (Projektschlüssel)
DATE	Erstellungsdatum der Tabelle
ABSOLUTE	(gewichtete) Fallzahl
PHYSICALRECORDS	(ungewichtete) Fallzahl
MIN <[ABSROW/ABSCOLUMN PHYSICALROW/PHYSICALCOLUMN]>	Minimalwert der ausgewählten Datenreihe
MAX <[ABSROW/ABSCOLUMN PHYSICALROW/PHYSICALCOLUMN]>	Maximalwert der ausgewählten Datenreihe
MEAN	globaler Tabellen-Mittelwert

Um auf die GESStabs-Elemente zugreifen zu können, müssen diese in der zugrundeliegenden Tabelle enthalten sein. Zum Einfügen von **ABSOLUTE** und **PHYSICALRECORDS** bzw. des jeweiligen Minimal-/ Maximalwertes sowie **MEAN** sind zudem globale Tabelleneinstellungen nötig: Das Ansprechen von Fallzahlen (**ABSOLUTE** und **PHYSICALRECORDS**) erfordert eine **TotalRow** und/ oder **TotalColumn**: **FRAMEELEMENTS = +TOTALROW +TOTALCOLUMN;** . Um den allgemeinen Mittelwert einer Tabelle (**MEAN**) anzeigen zu lassen, sind erstens beide genannten **FRAMEELEMENTS** erforderlich und zweitens die Ausgabe des Mittelwertes in den Datenzellen: **CELLELEMENTS = MEAN;** .

Den genannten Zahlen-Elementen können zudem optional Formatbeschreibungen hinzugefügt werden: **[FORMAT <"Formatbeschreibung">]**. Wird keine Formatbeschreibung angegeben, wird das für das jeweils geltende Zellelement geltende Format ausgegeben.

Die o.g. Inhalte können einzeln oder in Kombination (Verkettung) in Textboxen eingefüllt werden. Eine typische Syntax-Sequenz lautet etwa:

```
FILL TEXT_01 WITH CONCAT (TABLETITLE)
FILL TEXT_02 WITH CONCAT ("Dieser Text wird 1:1 übernommen")
FILL TEXT_03 WITH CONCAT (VARTITLE v1)
FILL TEXT_04 WITH CONCAT ("Basis = " ABSOLUTE FORMAT "#.##" ", Erstellungsdatum
: " DATE)
```

TEXT_04 enthält als Ergebnis folgenden Inhalt: “Basis = 1000.00, Erstellungsdatum: 04.11.2019“

START <Startzelle> markiert einen Startpunkt in der PowerPoint-Datentabelle, von dem aus die Datenbefüllung beginnt⁵.

Bei komplexen Charts kann es notwendig sein, Daten aus der GESStabs-Tabelle gezielt in genau adressierte Bereiche des mit dem PowerPoint-Element assoziierten Excel-Datenblattes zu

⁵Für ein Beispiel, siehe 3.6.

übertragen (siehe z.B. [Beispiel 6: START](#)). Mit dem **START**-Schlagwort legt man den Startpunkt dieses Zielbereichs fest und zwar wie folgt:

Für Grafiken: **START B2** (Excel-Zelle B2 im Datenblatt)

Für Tabellen: **START R2C2** (Reihe (**R**ow) 2, Spalte (**C**olumn) 2)

INVERSE dreht die Achsen des angesprochenen Datenbereichs⁶.

Dieses Schlagwort entspricht dem Excel-Kommando 'Transponieren'; Tabellenzeilen werden als Spalten, Spalten als Zeilen im PowerPoint-Element dargestellt.

Beachte: In PowerPoint-Tabellen verursacht **INVERSE** ein physisches Drehen der ausgewählten Datenbereiche, in PowerPoint-Diagrammen lediglich eine transponierte Darstellung der Datenreihen⁷.

NOLABEL [X | Y] bewirkt, dass Labeltexte in X-/ Y-Richtung nicht übertragen werden.

Standardmäßig werden die zugehörigen Labeltexte (oder wenn nicht vorhanden: Labelcodes) in die angesteuerten Elemente des PowerPoint-Charts übertragen. Wird eine in die Elemente integrierte Beschriftung nicht gewünscht, kann dies mit diesem Schlagwort unterdrückt werden.

UNITS <Text> ergänzt numerische Beschriftungen um frei wählbare Zeichen, z.B. %, €.

Sollen Labeltexte (grundsätzlich) nicht um **UNITs** ergänzt werden, kann dies in der Format-Angabe im **TABLE**-Statement eingestellt werden: **TABLE = Q1 by Q2 mean :format" '#.###^NOUNITS" ' SORT ABSOLUTE;**

In der Darstellung entfernt GESS ChartFactory die Angabe "**^NOUNITS**" und fügt keine **UNITs** an.

FIXEDRANGE und **FIXEDTABDIM** deaktivieren die automatische Dimensionierung von PowerPoint-Elementen.

Standardmäßig werden PowerPoint-Elemente an die Beschaffenheit der Daten angepasst: Diagramme und Tabellen werden entsprechend der gegebenen Variablen-, Label- und Kategorienanzahl ausgerichtet. Mit dem Schlagwort **FIXEDRANGE** kann dies für den aktuellen **POWERCHART**-Befehl unterdrückt werden: Unabhängig davon, wie viele Datenreihen in das PowerPoint-Element übertragen werden, behält dieses die in der Chartvorlage eingerichtete Größe des Datenbereichs bei und stellt diesen dar.

PowerPoint-Tabellen reagieren mit einer Veränderung ihrer äußeren Maße auf die übertragene Anzahl der Datenelemente. **FIXEDTABDIM** bewirkt, dass sich die Außenmaße der Tabelle nicht verändern. Dies ist z.B. nützlich, wenn eine Texttabelle als Beschriftung für ein Diagramm dient und die beiden Elemente in der Chartvorlage optisch aufeinander abgestimmt sind. **FIXEDTABDIMX** bzw. **FIXEDTABDIMY** bewirkt, dass nur die horizontalen bzw. vertikalen Maße einer PowerPoint-Tabelle fixiert werden.

DIRECTION [RISING | FALLING] steuert die Darstellungsrichtung der ausgewählten Daten.

Sollen Datenreihen entgegen der üblichen Darstellungsrichtung (von links nach rechts, von oben nach unten) dargestellt werden (siehe z.B. [Beispiel 7: DIRECTION FALLING](#)), kann die entgegengesetzte Darstellung mit **DIRECTION FALLING** bewirkt werden. Das Pendant ist **DIRECTION RISING**.

⁶Siehe [3.2](#) für ein Beispiel.

⁷Ein physisches Drehen der Datenreihen im Diagramm-Datenblatt wird mit [Transpon](#) bewirkt.

CHARTITLE "<string>" fügt einen frei wählbaren Text als Diagrammtitel (wenn in der Chart-Vorlage eingeblendet) ein.

** Text ** ist eine GESStabs-Anweisung, die in GESS ChartFactory zur Formatierung freier Texte in Textboxen eingesetzt werden kann.

Der Befehl hat folgende Bestandteile:

FONT ist das Befehlsschlagwort.

Es können optional Einstellungen für die Schriftart hinzugefügt werden: **BOLD** (fett), **UNDERLINE** (unterstrichen) und/ oder **ITALIC** (kursiv).

COLOR bewirkt eine Farbeanpassung (optional).

Es können sowohl RGB-Farbcodes (100,100,100) als auch Hexadezimal-Notationen (beginnend mit \$ oder #) angegeben werden.

SIZE verändert die (voreingestellte) Schriftgröße des Textelementes.

Optional kann eine Schriftgröße angegeben werden, z.B. **SIZE=12**, oder mit +/- die Schriftgröße in Relation zur Originalgröße des Textelements verändert werden, z.B. **SIZE=+3** und **SIZE=-3**.

Eine Sequenz lautet dann zum Beispiel:

```
FILL TEXT_01 with FREETEXT "<FONT ITALIC BOLD COLOR=255,0,0 SIZE=18>
Dieser Text wird 1:1 übernommen</FONT>"
```

MARKSTRING <Text> adressiert einen freien Text innerhalb eines TABLE_[xx]-Elements, der markiert werden soll⁸. Textteile können durch Wildcards ergänzt werden, siehe dazu [Wildcards in Suchtexten](#).

Dem MARKSTRING-Schlagwort folgt eine Spezifikation der gewünschten Formatierung. Beeinflusst werden können:

FOREGROUND <Farbcode>	Schriftfarbe, zu kennzeichnen mit Hexadezimal- oder RGB-Code
BACKGROUND <Farbcode>	Hintergrundfarbe, ebenfalls via Hexadezimal- oder RGB-Code
MARKBOLD	Fette Schrift

Durch die Erweiterung LINE innerhalb der Text-Definition kann eine ganze Tabellenzeile, die den definierten Text enthält, markiert werden. Wird innerhalb einer Tabelle eine Markierung vorgenommen, kann das gesamte Element mittels **Arrange** zudem optional in den Vordergrund (F=Front) oder Hintergrund (B=Back) verschoben werden.

Zum Beispiel: **MARKSTRING "Markiere diesen Textbereich,LINE,ARRANGE=F"**
BACKGROUND 128,128,128.

Sollen mehrere Texte via MARKSTRING markiert werden, kann dies als POWERCHARTOPTION [Markstring](#) umgesetzt werden.

Mit **POWERCHARTOPTIONS="Optionen"** können Diagramm- und Tabellen-Elementen in Power-Point anhand ihrer Inhalte (Texte und Zahlenwerte) optisch beeinflusst und verändert werden. Eine ausführliche Erklärung aller Optionen und der entsprechenden Syntax findet sich in [POWER-CHARTOPTION](#).

⁸Siehe hierzu auch [4.3](#).

HEATMAP <3digitcode> färbt die numerischen Werte in einer Tabelle als Heatmap ein.

Die erste Ziffer beschreibt die Methode, nach der die Datenzellen für die Färbung gruppiert werden:

1 = Standardnormalverteilung

2 = Perzentile

3 = gleiche Intervalle

Die zweite Ziffer legt die zu verwendende Farbpalette für die Markierung fest:

1 = Blau - Rot

2 = Blautöne

3 = Grüntöne

4 = Benutzerdefiniert: Im zweiten Datenblatt (namens 'Palette') der mit Excel geöffneten POWERCHART.xlsb kann die Farbpalette in Spalte H-K beliebig angepasst werden.

Die dritte Ziffer gibt an, ob die numerischen Werte in den Zellen sichtbar sein sollen:

0 = nein

1 = ja

Eine häufig genutzter Standardangabe ist 311.

11.3 Einfache Übernahme von Seiten

Eine reduzierte Form des POWERCHART-Befehls kann verwendet werden, wenn Seiten einer Vorlage-Datei 'blank' übernommen werden sollen, ohne PowerPoint-Elemente zu befüllen.

POWERCHART

INSERT = "<Vorlage-Datei>"

TEMPLATE <Nummer>

;

Eingesetzt werden können Seiten aus der zu Skriptanfang definierten oder einer anderen Vorlage-Datei. In jedem Fall muss hier deren Dateiname angegeben werden.

11.4 Zusätzliche Befehle

Die folgenden Statements werden **außerhalb des POWERCHART-Befehls** geschrieben.

POWERCHARTTEXTEXTRA [LABELS X | LABELS Y] = "[BOLD | UNDERLINE | SIZE = <Größenangabe>]"; macht globale Formateinstellungen für Labeltexte⁹.

Mit dem **POWERCHARTFORMAT**-Befehl kann die Übertragung von Informationen aus der GESSStabs-Tabelle in PowerPoint optional beeinflusst werden¹⁰. Wie alle übrigen Formatangaben in der GESS-Syntaxsprache, z.B. TABLEFORMAT, können die Optionen absolut oder relativ (+/-) benutzt werden. Die Einstellungen gelten für alle dem Statement folgenden Anweisungen.

⁹Siehe hierzu auch Kapitel 9.

¹⁰3.3 enthält ein Beispiel.

Die Syntax lautet `POWERCHARTFORMAT = Optionen;` und hat folgende Optionen:

AbsInLabelX	Ergänzung der gewichteten Spalten-Basen in den Labeltexten der X-Variable
AbsInLabelY	Ergänzung der gewichteten Zeilen-Basen in den Labeltexten der Y-Variable
PhysInLabelX	Ergänzung der ungewichteten Spalten-Basen in den Labeltexten der X-Variable
PhysInLabelY	Ergänzung der ungewichteten Zeilen-Basen in den Labeltexten der Y-Variable

`PPNUMFORMAT = '<formatstring>'` stellt das Format der Darstellung von Zahlen ein, die (Label-) Texten hinzugefügt werden.

Die standardmäßige Darstellung von Basen in Texten sieht einen Zeilenumbruch nach dem Labeltext und die Basenangabe in Klammern vor. Mit `PPNUMFORMAT = '<formatstring>';` kann das Zahlenformat beeinflusst werden. Der Standardformatstring lautet `|| (#)`, mit `PPNUMFORMAT = '(#)';` würde bei der Ausgabe der Basen beispielsweise kein Umbruch erzeugt.

`POWERCHARTREIMPORT = YES;` aktualisiert die in das Ergebnischart eingefügten Daten¹¹.

Nachträge Layout- oder andere Änderungen, beispielsweise an der Chartreihenfolge, bleiben bestehen und werden nicht überschrieben.

¹¹Siehe hierzu auch 8.

12 Syntax - Verzeichnis

Fett markierte Schlagworte müssen in lauffähigen Befehlen, in der angegebenen Reihenfolge enthalten sein.

Die übrigen Befehlselemente sind optional und zwischen GRATAB <PP-Name> und POSITION-Angabe frei positionierbar:

POWERCHART	Start des Charting-Prozesses
KEY <string>	Eindeutige Kennung für den POWERCHART-Befehl
TEMPLATE [<Nummer p>]	Vorlagechart: Seitennummer oder p=previous
<i>oder</i>	
TEMPLATE BY [ROWS COLUMNS] <{<count> : <Templatenr.>*n [ELSE <Templatenr.>]}>	Vorlagechart: dynamische Auswahl je nach Zeilen-/ Spaltenzahl der GESStabs-Tabelle
GRATAB <PP-Name>	zu bearbeitendes Element: GRAPH_[xx] oder TABLE_[xx]
START <Startzelle>	Startpunkt der Datenbefüllung in PP-Element
INVERSE	Gedrehte Darstellung des übertragenen Datenbereichs
NOLABEL [X Y]	Unterdrückte Übertragung der Labeltexte
UNITS <Text>	Ergänzung für numerische Beschriftungen
FIXEDTABDIM {X Y}	Fixierung der äußere Dimension einer PP-Tabelle
FIXEDRANGE	Fixierung der Datenreihen-Anzahl eines PP-Elements
DIRECTION [RISING FALLING]	Steuerung der Darstellungsrichtung
HEATMAP <3digitcode>	Einfärben numerischer Tabellenwerte zu einer Heatmap
CHARTTITLE "<string>"	Einfügen eines freien Textes als Diagrammtitel

FILL TEXT_[xx] with	Befüllen von Textfeldern:
CONCAT (<Element>*n)	
“Freier Text“ ’Freier Text’	freier Text, wird auf dem Chart abgebildet
VARTEXT <Variablenname>	Vartext einer Variable
VARTITLE <Variablenname>	Vartitle einer Variable
VALUELABELS [x/y] <Labelnummer>	Text eines spezifischen Labelcodes
VALUELABELS [x/y]	
POSITION <Nummer>	Labeltext an ausgewählter Tabellenposition
TABLERNAME	Tabellenname
TABLETITLE	Inhalt der Tabletitle-Box über der Tabelle
TABLEFILTER <Nummer>	Bezeichnung des Tabellenfilters
TOPTXT	Inhalt der oberen Textbox
BOTTOMTEXT	Inhalt der unteren Textbox
HEADER	Kopfzeile
FOOTER	Fußzeile
INSTITUTION	Name des Kunden (links unten an der Tabelle)
DOCUMENT	Dokumentenkenzeichnung (rechts unten an der Tabelle)
CONTENTKEY	Contentkey der Tabelle
GRAPHPROJECT	zusätzlicher Bezeichner (Projektschlüssel)
DATE	Erstellungsdatum der Tabelle
ABSOLUTE	(gewichtete) Fallzahl
PHYSICALRECORDS	(ungewichtete) Fallzahl
MIN <[ABSRW/ABSCOLUMN	
PHYSICALROW/PHYSICALCOLUMN]>	Minimalwert der ausgewählten Datenreihe
MAX <[ABSRW/ABSCOLUMN	
PHYSICALROW/PHYSICALCOLUMN]>	Maximalwert der ausgewählten Datenreihe
MEAN	globaler Tabellen-Mittelwert
<hr/>	
= ROWS <Labelcode> COLUMNS <	Angabe der zu übertragenden Labelcodes aus der Tabelle
Labelcode>	
oder	
POSITION = ROWS <Nummer>	Angabe des zu übertragenden Datenbereichs aus der Ta-
COLUMNS <Nummer>	belle
;	Abschluss eines Kommandos

Textformatierungen:

freie Texte in Textboxen und Tabellen

<FONT [BOLD/UNDERLINE/ITALIC] [COLOR=<Farbcode>] [SIZE=<Größenangabe>]>
Text

in Tabellen

MARKSTRING "<Text>{,LINE}{,ARRANGE=F|B}" {MARKBOLD} [FOREGROUND <Farbcode> |
BACKGROUND <Farbcode>]

POWERCHARTOPTION “Optionen“;

MARKSTRING=<Text{,LINE}{,ARRANGE=F|B}>{;MARKBOLD};
[FOREGROUND=<Farbcode>/BACKGROUND=<Farbcode>]

Transpon

LabelIconReplace=pos=L|C|R|{,resize=X|Y|B};<Text>=<Bild>

XYScatterIcon=pos=L|R|T|B|{,arrange=F|B};<Text>=<Bild>

BaseCheck=(Bedingung);{Remove|Italic|NoDataVal|
NoDataChar|NoLineFeed;}Bottomtext=<TEXT_[xx]>=
Hinweistext

Sort=<Tabellenspalte>{;ASCENDING}

ValueDiffTo [Max/Min]

Formel==Formel;Start=<Startzelle>
{;Range=<Tabellenbereich>}

DataFromGraph=GRAPH_[xx];<Zellenbereich>

Axis[Pr[i|2]/Se{c|2}]=<Skala-Min>;<Skala-Max>

CellColor=<Bedingung>;<Farbangabe>

CellFontcolor=<Bedingung>;<Farbangabe>

CellIcon=<Bedingung>;pos=L|C|R,resize=X|Y|B,
arrange=B|F,<Bild>

Per [Row/Col]

SeriesColor=<Bedingung>;<Farbangabe>

Markierung von Texten

physisches Drehen des übertragenen Datenbereichs

Ersetzen von Texten durch Bilder (in Tabellen)

Ersetzen von Texten durch Bilder (in X/Y-Diagrammen)

Markierung von Basen, nach Zahlenwert

Sortieren von Spalten-Werten (in Tabellen)

Neuberechnung der Zählwerte in Tabellen im Verhältnis zum kleinsten/ größten Wert

Einfügen von Formeln und Neuberechnung von Werten

Übernahme von Daten aus Diagramm-Datentabelle

manuelle Skalierung von Diagramm-Achsen

Farbige Markierung von Tabellenzellen nach Zahlenwert

Einfärben von Schriftfarben in Tabellenzellen, je nach Zahlenwert

Einfügen von Bildern in Tabellenzellen, je nach Zahlenwert

Anwendung der CellColor/CellIcon-Befehle für ganze Zeile/ Spalte (in Tabellen)

Farbige Markierung von Datenreihen, je nach Zahlenwert (in Diagrammen)

<code>SeriesLabelColor=<Bedingung>;<Farbangabe></code>	Farbige Markierung von Labeltexten, je nach Zahlenwert der Datenreihe (in Diagrammen)
<code>SeriesColorMarkstring=<Text>;<Farbangabe>{;Zahlenwert}</code>	Farbige Markerierung von Datenreihen, nach Labeltext (in Diagrammen)
<code>SeriesLabelAdd = <Ziffer> <Buchstabe></code>	Ergänzungen von Datenbeschriftungen (in Diagrammen)
<code>SeriesLabelReplace = <Ziffer> <Buchstabe></code>	Ersetzen von Datenbeschriftungen (in Diagrammen)
<code>FillColorObject=<PP-Element>;<Bedingung>;color=<Farbangabe>{;cell=[<row>,<column>]}</code>	Bedingtes Einfärben von PowerPoint-Objekten
<code>DeleteObject=<PP-Element>{;TEXT_[xx] <Bedingung>{;NDC}}</code>	(Bedingtes) Löschen von PowerPoint-Objekten

Zwischenblätter

POWERCHART

INSERT = "<Vorlage-Datei>"

TEMPLATE <Nummer>

;

Globale Formatierung für Labeltexte:

POWERCHARTTEXTEXTRA <[LABELS X | LABELS Y]> = <"[BOLD | UNDERLINE | SIZE = Größ
enangabe]">;

POWERCHARTFORMAT = [+ | -] Optionen;

AbsInLabelX	Ergänzung gewichteter Basen in Labeltexten auf X-Achse
AbsInLabelY	Ergänzung gewichteter Basen in Labeltexten auf Y-Achse
PhysInLabelX	Ergänzung ungewichteter Basen in Labeltexten auf X-Achse
PhysInLabelY	Ergänzung ungewichteter Basen in Labeltexten auf Y-Achse
PPNUMFORMAT = '<formatstring>';	Format der Zahlendarstellung
POWERCHARTREIMPORT = YES;	Datenaktualisierung

Stichwortverzeichnis

- AbsInLabelX/Y, 11, 85, 91
- Auswahlbereich, 4
- AutoWiederherstellen, 2, 77
- AXISPRI/SEC, 39, 58, 88

- BaseCheck, 51, 88

- CALCULATECOLUMN, 74
- Cellcolor, 59, 88
- Cellfontcolor, 60, 88
- Cellicon, 61, 88
- Charting starten, 3
- CHARTTITLE, 83, 87
- COMPUTE, 74

- DataFromGraph, 58, 88
- Datenblatt, 9
- DeleteObject, 69, 88
- Diagramm-interne Beschriftungen, 25, 75
- DIRECTION FALLING, 21, 83, 87

- FILL TEXT_[xx] WITH, 9, 26, 81, 87
- FillColorObject, 68, 88
- FIXEDRANGE, 83, 87
- FIXEDTABDIM, 29, 83, 87
- FIXEDTABDIMX, 83, 87
- FIXEDTABDIMY, 83, 87
- FONT, 27, 74, 84, 88
- Formula, 56, 88

- GESS-Produkte, 2
- GESStabs, 3, 8, 57, 73
- GRATAB, 9, 80, 87

- HEATMAP, 85, 87

- INSERT, 85, 91
- INVERSE, 10, 47, 83, 87

- KEY, 9, 79, 87
- Kontrolle von Formatierungen, 75

- LabelIconReplace, 27, 30, 48, 75, 88
- Liniendiagramm, 15, 33
- Lizenzierung, 1

- Makro, 36, 41
- Makro autochart, 43
- MARKSTRING, 29, 47, 84, 88
- Multiuser, 74

- NOLABEL X/Y, 14, 25, 83, 87
- NOZERODASH, 73, 76

- Per Row/Col, 62, 88
- PhysInLabelX/Y, 85, 91
- POSITION, 9, 80, 88
- POWERCHART, 9, 79, 87
- POWERCHARTFORMAT, 11, 25, 85
- POWERCHARTOPTION, 46, 84, 88
- POWERCHARTREIMPORT, 71, 86, 91
- POWERCHARTTEXTTEXT, 74, 75, 85, 91
- PowerPoint, 4, 75
- POWERPOINTFILENAME, 3
- PPEXCHANGE, 3
- PPNUMFORMAT, 11, 86, 91
- PPTEMPLATES, 3, 7
- Produktionsstraße, 2
- Profildiagramm, 38

- Schlüsselwörter, 5
- Seitenverhältnis sperren, 49, 75
- Seriescolor, 63, 88
- SeriescolorMarkstring, 65, 88
- SeriesLabelAdd, 67, 88
- SeriesLabelcolor, 64, 88
- SeriesLabelReplace, 67, 88
- Sort, 54, 88
- START, 17, 22, 35, 82, 87
- Syntax, 79, 87

- Tabelle, 12, 28, 75
- Tabelleninhalte, 73
- TABLE, 8, 79
- Technische Voraussetzungen, 2
- TEMPLATE, 9, 75, 80, 87
- TEMPLATE p, 12, 80, 87
- Textbox, 25, 75
- Texte, 25
- Transpon, 47, 88

- UNITS, 83, 87

- ValueDiffToMax/Min, 55, 88

- XLSB, 3
- XYScatterIcon, 50, 88

- Zwischenblatt, 85, 91